IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **ALAN NICHOLAS FLEET ET AL**	§	Attorney Docket No.: **TA-00418:0**
	§	
	§	
Serial No.: **09/595,550**	§	Examiner: **STEVENS, THOMAS H..**
	§	
Filed: **JUNE 16, 2000**	§	Art Unit: **2123**
	§	
For: **CASE BASED DRILLING KNOWLEDGE MANAGEMENT SYSTEM**	§	Confirmation No.: **1122**
	§	

Mail Stop OIPE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR REVIVAL OF AN APPLICATION FOR PATENT ABANDONED UNINTENTIONALLY UNDER 37 C.F.R. 1.137(b)

Dear Sir/Madam:

The above-identified application became abandoned on February 18, 2006, after the Applicant had paid the issue fee and believed that all informalities in this matter had been addressed. Accordingly, the Applicant submits this Petition for Revival of an Application and herewith:

(1) A reply required to the outstanding Office action or notice;

(2) A petition fee of $1,620.00; and

(3) A statement that the entire delay in filing the required reply from the due date for the reply until the filing of a grantable petition pursuant to this paragraph was unintentional.

Applicant notes that a terminal disclaimer (and fee as set forth in § 1.20(d)) is not required for the above titled application to be revived.

Exhibit 2

STATEMENT

Included with this petition is: (1) the petition fee of $1,620.00; (2) a copy of the executed declaration on file in the assignment division (Exhibit B and D); (3) a statement that the entire delay was unintentional; (4) a response to the outstanding action; (5) an amended specification to include an abstract and a description of the drawings; (6) a statement under 37 CFR 3.73(b) giving the assignee the power to act; and (7) a newly executed power of attorney from the assignee.

Applicant recently became aware of the abandonment while conducting an internal audit, and initiated an investigation into the circumstances surrounding the above titled application's abandonment. Applicant discovered that the original attorney of record, Melvin Hunn, was diagnosed with Leukemia and ultimately died during the pendency of this application, as shown by Exhibit A. His death on July 10, 2005 corresponds with the timing of the USPTO's attempts to correspond with the attorney regarding informalities issues in the application. It appears in the wake of Mr. Hunn's death, his partner Mr. Kenneth Hill paid the issue fee on Applicant's behalf. However, Mr. Hill failed to take heed of the USPTO's correspondence on Mr. Hunn's behalf regarding informalities, failed to notify the Applicant of such Office correspondence, failed to respond to a Supplemental Notice of Allowability sent after Applicant's payment of the issue fee, and failed to respond to or notify Applicant of the Notice of Abandonment.

Moreover, Applicant's investigation of the Examiner's action found that the Supplemental Notice of Allowability sent from the USPTO requiring the Applicant's submit an executed oath and declaration, and the record, revealed that the action leading to the abandonment of the above titled application was, at least partially, issued in error. An oath or declaration in this matter was transmitted to the USPTO on November 14, 2000, as shown by the file record. The file record of

the declaration does not contain a signature page, however. Applicant discovered the signature page for this declaration was recorded with the inventors' assignment of the above titled application on November 17, 2000, at Reel/Frame 011288/0676 (Exhibit B). The executed declaration is attached as Exhibit D. In this regard, Applicant also notes that the Office failed to change the name Alan Fleet to Alan Flett per the declaration and assignment papers. A publication submitted as patent literature showing the Inventor as Alan Flett is attached as Exhibit C.

Accordingly, Applicant requests, through its newly appointed counsel, to revive this application.

Because the attorney of record died, because the abandonment was at least partially the result of USPTO error, and because of the circumstances that led to the abandonment stem from the highly improbable circumstances that an issue fee would be paid for an application that still had formalities issues pending, Applicant hereby states that the abandonment, and entire delay between abandonment and this petition, was unintentional, and petitions for revival of this application in response to the notification of Abandonment issued.

A credit card payment is being made with this submittal for the above cited fees in the amount of $1,620.00. Again, since this application was filed on or after June 8, 1995, no terminal disclaimer is required.

Should there be any additional fees necessary for continued prosecution of this Application, the commissioner is hereby authorized to charge those fees to **Bracewell & Giuliani LLP's Deposit Account Number 50-0259 (9001RF.045342).**

Date: 8/1/11

Respectfully submitted,

Ashley L. Kirk
Registration No. 51,261
BRACEWELL & GIULIANI LLP
P.O. Box 61389
Houston, Texas 77208-1389
(713) 221-1544 direct phone
(713) 437-5381 direct facsimile
Email: ashley.kirk@bgllp.com
Attorney for Applicant

Thomas Miller, former Chronicle managing editor, dies at 66
posted on July 30, 2011 00:26:00 pm
Thomas C. Miller was a veteran newsman, teacher and former managing editor of the Houston Chronicle.

Singer Amy Winehouse dies at age 27 in London
posted on July 23, 2011 02:32:00 pm
Amy Winehouse, the beehived soul-jazz diva whose self-destructive habits overshadowed a distinctive musical talent, was found dead Saturday in her London

Funeral Home Details

> Website
> Map/Directions

Greenwood, Mount Olivet and Arlington Funeral Homes
Fort Worth and Arlington, TX

Helpful Resources

Send Flowers & Gifts

The FTD® *Comfort Planter*

$41.40 send now

In partnership with FTD >

> View More Arrangements > Visit Gift Shop

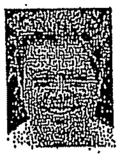
Helpful Links

> Donate to charity
> A Printed Guest Book is Available

Exhibit A

Obituary

Melvin Albert Hunn | Visit Guest Book



MELVIN ALBERT HUNN FORT WORTH -- Melvin Albert Hunn, 46, passed from this life Sunday, July 10, 2005, after a long and courageous battle with leukemia. Memorial service: 12 p.m. Wednesday at St. Patrick Cathedral, Fort Worth, with a reception following. Memorials: In lieu of flowers, donations may be made in Mel's honor to the Leukemia and Lymphoma Society or to Harris Methodist Hospital, Fort Worth. Melvin was born Nov. 3, 1958, in Sancti Spiritus, Cuba. His parents fled Cuba with him to escape the communist revolution. He grew up in Midland and graduated in 1977 from Robert E. Lee High School. He received his Bachelors of Science in Electrical Engineering in 1981 from the University of Texas at Austin and his Doctorate of Jurisprudence from Texas Tech University School of Law in 1986. Admitted to the Texas State Bar in November 1986, Mel earned his patent license in 1987 and his Federal Circuit License in 1992. For 19 years Melvin practiced intellectual property law in Fort Worth, commencing with Felsman, Bradley, Gunter & Dillon where he made partner in 1990. Melvin was on the board for the Young Lawyers Association of Fort Worth in the late 1980s, and in 2000 was named one of the top intellectual property attorneys by Fort Worth magazine. That year Melvin, along with his law partner and friend, Ken Hill, started the partnership of Hill & Hunn, LLP. In 2003, a prototype of a patent he wrote for a client was accepted by the Smithsonian Institution. He was greatly respected in the legal profession, and he was much appreciated by his clients. Melvin, also known as "Mel" or "Melle Mel" to those closest to him, loved life, art, jazz, restaurants and people. In 1992, along with three partners, he opened the January Gallery on 7th Street where he was able to indulge his love for avant garde art. Known for his great wit and insight, Mel continually entertained his extended family and numerous friends with clever quips and deep thought. He touched many lives with his compassion, understanding and sage advice, as Mel was an advisor, mentor, and confidante to countless friends and family members. Survivors: Beloved son, Clayton Anthony Hunn; Clay's mother, June Scobey of Arlington; parents, Terry and Alicia Hunn of Round Rock; younger siblings, Clay, Martha, Alan, Richard and Trina; Mel leaves behind many friends and relatives, including a host of "adoptive" brothers, sisters, nieces, nephews, sons and daughters. He looked forward to marrying his much-loved fiancée, Suzy Bradshaw, and being a step-father to her daughter Shannon. Greenwood Funeral Home 3100 White Settlement Fort Worth TX 76107 817-336-0584

Published in Houston Chronicle on July 19, 2005

- Follow this Obituary
- Email to a friend
- Share on Facebook
- Create a Bookmark
- Post on Other Sites
- Print this Obituary

Recommend

Guest Book

Melvin Hunn Guest Book | View 2 of 19 Entries:

"Melvin was a piece of art and invention in his own right. After initiating a patent for me, my daughter I met with Melvin and his son Clay for some brain storming ideas.My patent idea would involve..." - Kathleen Lieberman

"A fine friend and colleague." - Guy Manning

View Additional entries from:

- Robin Collins-Morris
- Justine Vanthill (Ithaca, NY)
- Eddie and Hilda Graham (Houston, TX)

View Complete List

Restore the Guest Book now

Related Memorial Sites ▪ University of Texas

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

April 25, 2011

THIS IS TO CERTIFY THAT ANNEXED IS A TRUE COPY FROM THE RECORDS OF THIS OFFICE OF A DOCUMENT RECORDED ON NOVEMBER 17, 2000.

By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office

R. BLAKENEY
Certifying Officer

Exhibit B

12-05-2000

101538529

11-17-00

DOCKET NO. 9001RF-34875 RECORDATION FOR[illegible] ...RCE

Form PTO-1595

Patent and Trademark Office

To the Honorable Commissioner of Patent... ...ached original documents or copy thereof.

OIPE NOV 1 7 2000 JC98 PATENT & TRADEMARK OFFICE

1. Name of conveying party(ies):

Alan Nicholas Flett
Derek H. Sleeman
Alun D. Preece

Additional name(s) of conveying party(ies) attached? ☐ Yes ☒ No

2. Name and address of receiving party(ies):

Name: Baker Hughes Incorporated

Internal Address: P.O. Box 4740
Houston, Texas 77210-4740

Street Address: 3900 Essex Lane

City: Houston State: TX ZIP: 77027

Additional name(s) & address(es) attached? ☐ Yes ☒ No

3. Nature of conveyance:

☒ Assignment ☐ Merger
☐ Security Assignment ☐ Change of Name
☐ Other

Execution Date: 08/21/00; 09/03/00; 09/01/00

4. Application number(s) or patent number(s):

If this document is being filed together with a new application, the execution date of the application is:

A. Patent Application No.(s)
09/595,550

B. Patent No.(s)

Additional numbers attached? ☐ Yes ☒ No

5. Name and address of party to whom correspondence concerning document should be mailed:

Name: Melvin A. Hunn

Internal Address: HILL & HUNN LLP
201 Main Street, Suite 1440
Fort Worth, Texas 76102-3105

Street Address: HILL & HUNN LLP
201 Main Street, Suite 1440

11/30/2000 JADDO1 00000072 020429 09595550
01 FC:581 40.00 CH

City: Fort Worth State: Texas ZIP: 76102-3105

6. Total number of applications and patents involved: 1

7. Total fee (37 CFR 1.21(h)..........$ 40.00

☐ Enclosed

☒ Authorized to be charged to deposit account

8. Deposit Account Number:

02-0429

(Attach duplicate copy of this page if paying by deposit account)

DO NOT USE THIS SPACE

9. Statement and signature.

To the best of my knowledge and belief, the foregoing information is true and correct and any attached copy is a true copy of the original document.

Suzy Bradshaw
Name of Person Signing

Signature

14 NOV 00
Date

PATENT APPLICATION ASSIGNMENT

WHEREAS, WE, **Alan Nicholas Flett**, having a correspondence address of 8 Mill ade Wynd, Danestone, Aberdeen AB22 8QN, Scotland; **Derek H. Sleeman**, having a correspondence address of 10 The Chanonry, Aberdeen AB24 1RN, Scotland; and **Alun D. Preece**, having a correspondence address of 10 Fare Park Drive, Westhill, Aberdeenshire AB32 6WE, Scotland have invented new and useful improvements in a

CASE BASED DRILLING KNOWLEDGE MANAGEMENT SYSTEM

for which application is being made for Letters Patent, said application further identified as U.S. Patent Application Serial No. **09/595,550**, filed **16 JUNE 2000**.

WHEREAS, **BAKER HUGHES INCORPORATED**, a corporation of the State of Delaware, having a correspondence address at P.O. Box 4740, Houston, Texas 77210-4740, is desirous of acquiring our entire right, title and interest in said application and any Letters Patent which may issue thereon:

NOW THEREFORE, be it known by all whom it may concern that for and in consideration of One Dollar ($1.00), the receipt of which is hereby acknowledged, and other good and valuable consideration, I hereby assign to said corporation for the territory of the United States of America and the entire world our entire right, title and interest in and to said invention, patent application and patent, including all priority rights for patent applications foreign to the United States of America.

I HEREBY covenant and agree that I will at any time, upon the request and at the expense of said corporation, execute and deliver any and all papers and do all lawful acts that may be necessary or desirable to perfect the title to said invention, applications and patents, and I authorize the Commissioner of Patents and Trademarks to issue Letters Patent to said corporation.

IN TESTIMONY WHEREOF, I execute this assignment on the date set forth below my name.

Date: 21/8/2000 — Alan Nicholas Flett

Date: 3 Sept 2000 — Derek H. Sleeman

Date: 1/9/2000 — Alun D. Preece

Fort Worth/9001RF-31037/67032.1

Better Knowledge Management through Knowledge Engineering

Alun Preece, Alan Flett, and Derek Sleeman, *University of Aberdeen*
David Curry, Nigel Meany, and Phil Perry, *Baker Hughes OASIS*

Currently, few organizations have a systematic process for capturing knowledge, as distinct from data. The authors illustrate how a large oil and gas service company uses knowledge-engineering processes to capture, store, and deploy drilling-optimization knowledge.

In recent years, *knowledge management* has referred to efforts to capture, store, and deploy knowledge using a combination of information technology and business processes.[1–3] More specifically, organizations aim to acquire knowledge from valued individuals and to analyze business activities to learn from successes and failures. Such captured knowledge must then be made available throughout the organization in a timely manner.

In terms of technology, most current knowledge management activities rely on database and Internet systems. If knowledge is stored explicitly at all, it is typically in databases either as simple tables (for example, relational databases) or semistructured text (as in Lotus Notes). The use of sophisticated knowledge representation systems such as Classic, Loom, or G2 is rare. Also, few organizations have a systematic process for capturing knowledge, as distinct from capturing information. (See the "Current Practice" sidebar for a description of techniques.)

We believe that current knowledge management practice significantly under-utilizes knowledge-engineering technology, despite recent efforts to promote its use.[4] In this article, we focus on two knowledge-engineering processes:

- using knowledge acquisition processes to capture structured knowledge systematically and
- using knowledge representation technology to store the knowledge, preserving important relationships that are far richer than those possible in conventional databases.

To demonstrate the usefulness of these processes, we present a case study in which the drilling optimization group of a large oil and gas service company uses knowledge-engineering practices to support the three facets of the knowledge management task:

- *Knowledge capture*—In the group's systematic knowledge acquisition process, a conceptual business model of the company guides case and rule capture.
- *Knowledge storage*—The group uses a knowledge representation language to codify the structured knowledge in several knowledge bases, which together make up a knowledge repository.
- *Knowledge deployment*—Through standard Web browsers on the company intranet, group members can run the knowledge bases within a knowledge server. The server answers queries far more complex than those possible with conventional database systems.

Applying knowledge engineering to knowledge management

In the 1990s, knowledge engineering emerged as a mature field, distinct from but closely related to software engineering.[3,5] Among its distinct aspects are a range of techniques for knowledge elicitation and modeling, a collection of formalisms for representing knowledge, and a toolkit of mechanisms for implementing automated reasoning.

1094-7167/01/$10.00 © 2001 IEEE

Exhibit C

Here is an outline of the knowledge-engineering process:[3,6]

1. *Requirements analysis*. Identify the scope of the knowledge-based system, typically in terms of its expected competency (for example, the kinds of queries it will be able to answer).
2. *Conceptual modeling*. Based on the scope defined in step 1, create a glossary of terminology (concepts) for the application domain and define interrelationships between the terms of and constraints on their usage. An explicit conceptual model of this kind is commonly called an *ontology*.
3. *Knowledge base construction*. Using the conceptual model or ontology from step 2 as a collection of *knowledge containers* (or schemata), populate the knowledge base with *instances* of domain knowledge (often in the form of rules, facts, cases, or constraints).
4. *Operationalization and validation*. Operationalize the knowledge base from step 3 using automated reasoning mechanisms and validate its competence against the requirements from step 1. If satisfactory, release the system; otherwise, repeat steps 1 through 4 until satisfactory.
5. *Refinement and maintenance*. After delivery, the system continues to evolve as knowledge changes. Thus, steps 1 through 4 must be repeated throughout the life of the system.

Any knowledge management system that involves explicit knowledge representation is amenable to development using at least part of this process. In fact, it is always worth applying at least part of this process to any knowledge management activity that involves explicit knowledge representation. Here are several examples, using the common knowledge management activities described in the "Current Practice" sidebar:

- *Document management systems*. As a minimum, apply step 1 at the outset to ensure competency criteria are defined. This ensures at least the selection of the right tool; it may reveal a need for a more structured approach.
- *Discussion forums*. As a minimum, apply steps 1 and 2 to ensure that the system's scope is well understood, and that each forum's organization effectively supports existing (or desired) communities of practice.
- *Capability management systems*. As above, apply steps 1 and 2 to define the metaknowledge that will serve as knowledge containers or schemata to capture workers' capabilities. Use step 3 to populate the CV database.
- *Lessons-learned knowledge base systems*. Because these are knowledge-based systems, they should follow the entire five-stage process.

It is particularly important to employ knowledge-engineering techniques when an organization employs a range of knowledge management approaches. This is becoming common in larger organizations, which already use a multiplicity of information systems tied into an intranet and see a multifaceted knowledge management system as normal. For example, such a knowledge management system might include a capability management system, discussion forums, a document management system, and several lessons-learned knowledge bases. In such cases, the key challenge becomes *knowledge integration*—linking the various sources at the knowledge-content level.

In this context, the organization can use the knowledge-engineering process to define an organizational knowledge model—a *knowledge map*[7]—which delineates the relationships that bind the multifaceted knowledge management system at the knowledge-content level. (The actual software-level bindings can use hyperlinking, remote procedure calling, or any one of a host of distributed computing techniques.) Therefore, even when an organization embarks on its first, single-facet knowledge management project, it may well be worthwhile to follow steps 1 and 2 of the knowledge-engineering process to define an initial knowledge map.

Case study: drilling optimization

Baker Hughes OASIS, an engineering services subsidiary company of Baker Hughes, provides drilling-process expertise in the oil and gas industry worldwide. In particular,

Current Practice

Most knowledge management activities combine business processes and information technology.[1] As currently practiced, knowledge management includes several activities and technologies:

- *Document management systems* allow workers to find existing documents relevant to the task at hand. Essentially, these are multisource search and information-retrieval systems that tie into an organization's intranet (and may extend to the public Internet). These systems include several commercially available products, such as those made by Autonomy and Verity.
- *Discussion forum systems* promote knowledge dissemination within communities of practice. Workers subscribe to forums relevant to their interests, exchanging questions and answers, lessons learned, announcements, and industry gossip. Such systems are easily implementable with both freely available Web software and commercial products.
- *Capability management systems* allow an organization to "know who knows what."[2] Essentially, these are databases of suitably structured CVs or resumes; as such, they are implementable with off-the-shelf database software. The goal is to put people together by matching one person's need for expertise with another person's listed skills.
- *Lessons-learned knowledge base systems* let workers tap into past experience, by storing that experience as structured cases. These systems allow sophisticated queries, typically supporting "fuzzy" retrieval of "similar" cases. Although simple systems can use just conventional database software, full functionality requires special-purpose, case-based reasoning or knowledge-based system software.

References

1. W. Bukowitz and R. Williams, *Knowledge Management Fieldbook*, Prentice-Hall, Old Tappan, N.J., 1999.
2. J. Stader and A. Macintosh, "Capability Modeling and Knowledge Management," *Applications and Innovations in Intelligent Systems VII*, Springer-Verlag, Berlin, 1999, pp. 33–50.

Baker Hughes OASIS specializes in *drilling performance optimization*, which involves identifying, understanding, and overcoming barriers to improved drilling performance. Drilling performance optimization engineers need a specialized set of skills, which they draw from mechanical engineering, geology, physics, and other disciplines. Because the field is relatively new, the community of skilled optimization engineers is small, and those within Baker Hughes OASIS are dispersed worldwide.

For these reasons, drilling performance optimization represents an ideal application domain for knowledge management. Having recognized this in the early 1990s, Baker Hughes OASIS developed a multifaceted knowledge management approach, which currently includes the following system components:

- *Drilling Performance Guidelines*, a semistructured document base implemented in Lotus Notes/Domino;[8]
- *OASIS University*, an online training system for optimization engineers, also implemented in Lotus Notes/Domino;
- *Drill Bit Advisor*, a rule-based expert system implemented in LISP/CLOS using a custom graphical rule representation;[9] and
- *Drilling Knowledge Store*, a technical lessons-learned knowledge base.

All of these components are interlinked. For example, a conclusion (recommendation) made by the Drill Bit Advisor is commonly linked with a URL to a Drilling Performance Guideline in the Lotus Notes/Domino system.

The Drilling Knowledge Store, one of the newest components of this knowledge management strategy, is an open repository of case-based drilling knowledge, accessed through a Lotus Domino server. A structured search tool allows users to query the knowledge store for lessons learned in environments similar to a specified environment of interest. New knowledge forms promote easy entry of new cases, which the system submits to reviewers for audit and approval before making them available to other users. Links to the Drilling Performance Guidelines system avoid knowledge duplication and ease updating and maintenance.

The Drilling Knowledge Store builds on a knowledge map developed using the standard knowledge-engineering process described earlier, and it incorporates a drilling knowledge repository, a case-base of optimization engineers' documented experience. The drilling optimization group developed this case-base in collaboration with the University of Aberdeen, managing the work as a Teaching Company Scheme. The following sections detail its development stages.

Requirements analysis

The development team first conducted a series of interviews with optimization engineers to explore the scope of the drilling knowledge repository. The key finding was that the system ought to be highly open. Because drilling optimization is relatively new, knowledge in the domain is evolving. As a result, the system would most likely have to cope with the following kinds of change:

> By the time the team had defined a reasonably complete conceptual model, they had already elicited several sample cases from the optimization engineers as a natural part of exploring the scope of the domain.

- New concepts and relationships could be discovered in the future, so knowledge containers or schemata would have to be highly extensible.
- New cases would grow in proportion to the growth in the drilling optimization business, so instances would frequently be added.
- Instances might be reclassified, especially as outdated knowledge is "decommissioned."

Conceptual modeling

Following the first round of interviewing, the development team drew up an initial glossary of terms. In an attempt to derive a set of concepts, the team analyzed the transcripts of the interviews using the PC-PACK[4] knowledge acquisition software toolkit. However, it was not sufficiently flexible in dealing with concepts where the "defining" words were not adjacent in a piece of text or where they were interspersed with words from other concepts. PC-PACK and similar textual mark-up systems allow the user to indicate only that single words correspond to concepts, attributes, and values. In practice, such entities are often defined by several words, and these are not necessarily adjacent. For example, the text "a bus system that links all the suburbs to the center and to each other" contains the concept *comprehensive-city-bus-network*, but it also contains parts of the concept *city* (suburb and center).

In view of this tool's limitations, the team used a manual concept-mapping approach instead,[10] which focused on defining concepts in two areas:

- concepts associated with the drilling environment, including extensive definitions of geological concepts (leading to the creation of an ontology for representing the rock formations that constitute a drilling task), and those associated with drilling itself (chiefly drill bits, fluids, and related apparatus); and
- knowledge management concepts that would allow the capture of useful instances of the optimization engineers' experience (most obviously, the concept of a *case*).

Early in the process, the team formalized these concepts to manage them within a software environment. They chose the Loom knowledge representation system[11] and its associated Ontosaurus browser/editor because it had a number of advantages. First, Loom is one of the most flexible and least constraining knowledge representation systems available. In addition, Loom's operational mechanisms (chiefly the classification engine) allowed the knowledge-engineering team to test the conceptual model's integrity during its development. Finally, Ontosaurus provides a Web front end for Loom knowledge bases, allowing multiple users to inspect, query, and modify the knowledge base on a network, using a standard Web browser.

Knowledge base construction

By the time the knowledge-engineering team had defined a reasonably complete conceptual model, they had already elicited several sample cases from the optimization engineers as a natural part of exploring the scope of the domain. When formalizing the conceptual model in Loom, the team took the opportunity to represent these cases using

Loom's knowledge containers. However, they used a distinct approach for systematic case acquisition.

First, they identified a small number of high-performing, expert optimization engineers. Then, team members conducted intensive, one-on-one *knowledge acquisition campaigns* with these individuals. Mindful of lessons learned from negative knowledge acquisition experiences during the heyday of expert systems in the 1980s, the team carefully designed these campaigns to ensure that the experts would contribute actively and positively.

The team formalized the knowledge acquired from each campaign in Loom, but also wrote it up in a natural-language electronic document called a *knowledge book*,[12] which the expert could check for accuracy and which was disseminated on CD-ROM throughout the company as an easily accessible, early result of the OASIS group's work.

Operationalization and validation

The knowledge base was operationalized naturally by the choice of Loom as the representation language. The team performed validation of the represented knowledge at two levels—indirect validation using the knowledge books and direct validation using Loom's inference mechanisms. Further indirect validation came through the development of drill bit selection rules using some of the case-based knowledge acquired in the knowledge acquisition campaigns. These rules were then validated using existing software in the Drill Bit Advisor expert system.

The drilling knowledge repository in Loom

Philosophically, we consider OASIS a knowledge storage and retrieval system rather than a knowledge-based system. This is because knowledge-based systems are strongly associated with tasks, such as decision support, automated decision-making, or training. But the task for which knowledge is to be used places a strong bias on the form of the knowledge.[13] In this case, the implemented system had to be task-neutral: it was to serve purely as a repository for captured knowledge, without any risk of biasing the form and content of the knowledge toward a particular future usage.

Nevertheless, the system does have at its core a set of automatic deductive facilities (provided by Loom), which operate on the definitions given to the system by the knowledge modeler. However, these inferences operate at the conceptual-model level and not at any action level. Thus, actions are left to humans, and the system does not advise per se on any action the user should take. For example, the system can recognize instances and classify them appropriately with reference to the conceptual model, but this is purely for the purpose of retrieving those instances and bringing them to the user's attention.

The Loom knowledge store has two main parts—a conceptual model and a database. (This is analogous to a database, with its schema and data parts.) The conceptual part of the knowledge base is defined using concepts. It includes binary concepts (known as *roles*) and unary concepts (known as *concepts*). The database is populated with

The system does have at its core a set of automatic deductive facilities (provided by Loom), which operate on the definitions given to the system by the knowledge modeler.

instances of these concepts.

The following sections give examples of the Loom constructs to illustrate the approach in concrete terms. Our intention is to explain the constructs so that a full understanding of the representation language is not necessary. (For readers unfamiliar with Loom or similar languages, Robert MacGregor and Ronald J. Brachman and his colleagues provide good introductions.[11,14])

Modeling constructs for drilling engineers' experience

Because the knowledge store is chiefly intended to capture experiential cases from drilling engineers, the most important concept is the *case*.

```
(defconcept CASE :is-primitive
    (:and (:exactly 1 formation-sequence)
        (:all decision DECISION)
        (:all observation OBSERVATION)))
```

A case usually describes a *drill bit run*—a continuous period of drilling with a single drill bit. So, if an optimization engineer experiences some bit run worthy of being recorded in the knowledge store, the engineer should include a representation of the rock formation sequence and the decisions made on how to drill that formation sequence, along with any associated observations. A decision can refer to a choice of drill bit, mud (drilling fluid), flow rate, and so on. Alternatively, the case need not refer to an actual drill bit run if the person entering it simply has an experience to share.

A *decision* has several different dimensions, including *issues*, *actions*, *goals*, an author, a *spin*, and *reasoning*. These dimensions provide a balance between structured knowledge and free text. The structured knowledge enables formal representation and therefore supports powerful searches; the free text supports semistructured knowledge.

```
(defconcept DECISION :is-primitive
    (:and (:exactly 1 action)
        (:at-most 10 issue)
        (:at-most 10 goal)
        (:at-most 1 authors-reasoning)
        (:at-most 1 companys-reasoning)
        (:at-most 1 author)
        (:at-most 1 spin)))
```

An *issue* is some informational context that the engineer considered when making the decision. The issues in the current knowledge base reflect quite strongly the best-practice drilling database (in Lotus Notes), as shown by the link roles in the following code. These can be filled with links to other media, including the Notes database itself, using URLs.

```
(defconcept ISSUE :is-primitive
    (:and KNOWLEDGE_MANAGEMENT_CONCEPT
        (:at-most 1 symptoms-and-diagnosis-link)
        (:at-most 1 description-link)
        (:at-most 1 parameters-link)
        (:at-most 1 diagnostic-information-link)
        (:at-most 1 planning-actions-link)
        (:at-most 1 operating-practices-link)
        (:at-most 1 examples-link)))
```

An *action* is the real-world consequent the engineer performed as part of the decision; this includes both structured (categorical-outcome) and free text (textual-outcome) outcomes.

```
(defconcept ACTION :is-primitive
    (:and KNOWLEDGE_MANAGEMENT_CONCEPT
        (:at-most 1 categorical-outcome)
        (:at-most 1 textual-outcome)))
```

The system captures two kinds of *reasoning* for a decision. *The author's reasoning* is a free-text field for explanations—for example, why an engineer chose a certain drill bit. This allows the storage of incomplete, inaccurate, and even incoherent explanations for actions. After all, the main reasoning or determinism for the action consists of the other structured information describing the circumstances for the action, such as the formation sequence. The *company's reasoning* field expresses the company's commonly agreed on beliefs for the decision in question.

Modeling constructs for the drilling environment

The system describes the drilling environment chiefly in terms of conceptual rock sequences. The team achieved representations of these by defining an ontology of geological concepts, including constraints. For instance, if the user wishes to specify the depth or length of a particular section of *lithology* (a basic rock type—for example, sand or shale), that section must be represented as a *formation*. The superstructure larger than that is the *formation sequence*, which can have one or more *formations*. Each formation can have one or more *lithologies*. A formation is the conceptual modeling granularity at which the users should represent any part of the wells they feel should have represented interval lengths and depths.

```
(defconcept FORMATION_SEQUENCE :is-primitive
    (:and ROCK_CONCEPT
        (:at-least 1 formation)))

(defconcept FORMATION :is-primitive
    (:and ROCK_CONCEPT
        (:at-least 1 lithology)))
```

To allow users to represent and query formation sequences flexibly, the ontology defines several relations. For example, the relation comes-in-somewhere-after relates two formations, the first of which comes in somewhere after the other.

```
(defrelation comes-in-somewhere-after
  :domain FORMATION_SEQUENCE
  :range FORMATION
  :characteristics (:multiple-valued :closed-world)
  :is (:satisfies (?formation-x ?formation-y)
    (:and
      (FORMATION ?formation-x)
      (FORMATION ?formation-y)
      (:or (comes-in-immediately-after
          ?formation-x ?formation-y)
        (:exists (?formation-z)
          (:and
            (FORMATION ?formation-z)
            (comes-in-somewhere-after
              ?formation-x ?formation-z)
            (comes-in-somewhere-after ?formation-z
              ?formation-y)))))))
```

One important feature of lithologies is their *hardness*. While a lithology has, by definition, one rock type (such as shale), it can have more than one hardness. (For example, shale could consist of 100 meters of *very soft rock* and 300 meters of *soft rock*.)

The system describes the drilling environment chiefly in terms of conceptual rock sequences. The team achieved representations of these by defining an ontology of geological concepts.

```
(defrelation hardness
    :domain LITHOLOGY
    :range HARDNESS
    :characteristics (:closed-world :multiple-valued))
```

To support drill bit run modeling, the ontology includes a collection of functions that relate formation sequences, constituent lithologies, and accumulated hardness.

In addition to the generic geological concepts, the knowledge store includes representations of the concepts involved in drilling, such as *drill bit*.

```
(defconcept DRILL_BIT :is-primitive
    (:and DOWN-HOLE_EQUIPMENT_CONCEPT
        (:exactly 1 bit-gauge)))
```

Querying the knowledge store

The retrieve function, which retrieves instances from the knowledge base, provides an interface to Loom's deductive query facility. Formation sequence queries are among the most sophisticated forms of query that users can issue to the knowledge store. The concepts likely to be of interest are individual formations and formation sequences. Two common queries are on an overall cumulative amount of a certain hardness of a particular lithology over a formation sequence and formations that have amounts of particular lithologies of a certain hardness.

The following example query looks for cases that have a formation sequence that has as constituents of its formation(s) at least 1,900 feet of very soft to soft shale (including all subtypes of shale).

```
(retrieve ?case
  (:and
    (CASE ?case)
    (>= (sum (:collect ?lithology-amount-ft
      (:and
        (:exists (?formation-sequence ?formation
            ?lithology ?hardness)
          (:and
            (formation-sequence ?case ?formation-
              sequence)
            (formation ?formation-sequence
              ?formation)
            (lithology ?formation ?lithology)
            (lithology-hardness-amount-ft ?lithology
              ?hardness ?lithology-amount-ft)
            (:or
              (VERY_SOFT ?hardness)
              (SOFT ?hardness))
            (SHALE ?lithology)
            ))))) 1900)))
```

Users typically also want to look for cases in which engineers achieved specific goals or outcomes. The following example query retrieves cases that have a drill bit decision in which one of its goals was *good ROP* (rate of penetration) with *good bit cleaning*.

```
(retrieve ?case
  (:and
    (CASE ?case)
    (:exists (?decision)
      (decision ?case ?decision)
      (DRILL_BIT_PLANNING_DECISION ?decision)
      (goal ?decision
GOOD_ROP_WITH_GOOD_BIT_CLEANING))))
```

It is worth emphasizing how the Loom representation supports querying. First, the classification engine automatically associates new concepts (including new cases) with

super- and subconcepts. This means that a query for a subconcept will automatically find all superconcepts. This also means that, by using the Ontosaurus concept browser, a user can quickly find subconcepts related to the result of a query. Second, the pattern-matching mechanism, combined with the way Loom represents drilling sequences, means that the system easily accommodates partial matches. A user need only specify discontinuous fragments of a formation sequence, for example, to retrieve useful cases of drilling wells including those sequence fragments.

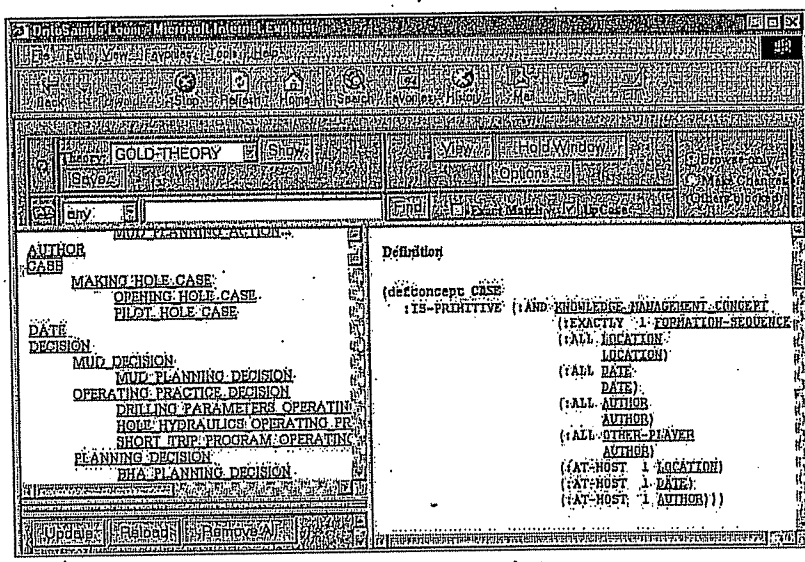


Figure 1. Loom Drilling Knowledge Repository screenshot.

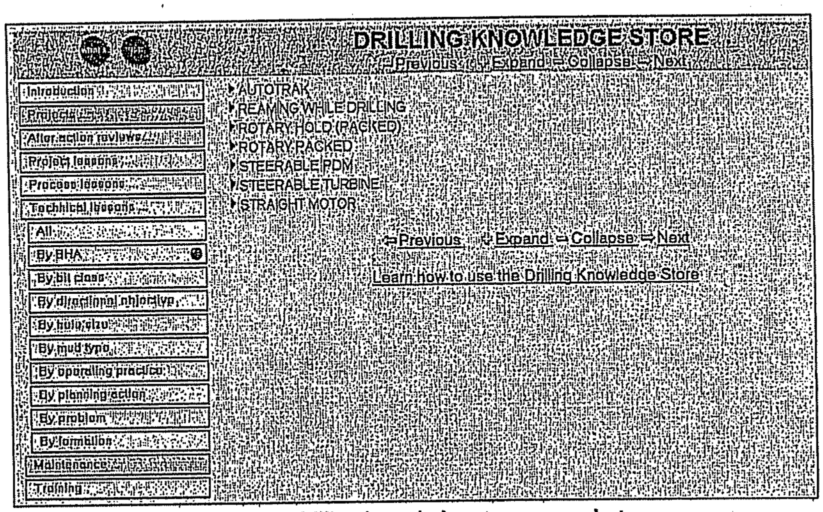


Figure 2. Lotus Notes/Domino drilling knowledge store screenshot.

Adding to the knowledge store

As we described earlier, the knowledge store comprises a conceptual and a database part. We consider the conceptual part stable and expect that knowledge will rarely need to be added or modified. However, additions to the database part will surely be regular. The Loom operations used to update the database part of the knowledge base are `tell` and `about`; `tell` is used to assert propositions and facts about the world or domain; `about` references the instance to which those propositions refer. The following example shows how a user might enter a case instance. This example case has one formation sequence name and zero or more decisions and observations.

```
(tell (:about Case-Name
    CASE
    (formation-sequence Formation-Sequence-
        Name)
    (decision Decision-Name)
    (observation Observation-Name)))
```

Current status and future plans

The Loom Drilling Knowledge Repository currently contains 1,200 concepts and 240 relations, with further expansion planned. The knowledge store is accessible on the company's intranet using a standard Web browser through the Ontosaurus system (see Figure 1).

The use of Loom has facilitated great flexibility in the modeling process allowing the ontology to grow naturally over the first two years of the project. Attempting to model a comparable richness of interrelationships in a relational database, for example, would have been extremely difficult and more time-consuming, and it would doubtless have involved many more modifications to the schemas.

Nevertheless, although it is relatively straightforward to browse the case base and ontology using Ontosaurus, the current system has several significant problems:

- It is difficult to issue complex queries to the Loom knowledge store through the Ontosaurus interface because Ontosaurus provides direct support only for simple queries (retrieve cases with matching simple role values).
- It is hard to add new cases because these require the user to have knowledge of Loom syntax, an unrealistic expectation for optimization engineers.
- Multiuser access (basic locking and restricted concurrent access) is limited.

In addition to these issues, the team wanted the drilling knowledge repository to have a familiar interface, preferably that of the existing systems implemented using Lotus Notes/Domino. At the same time, the team wanted to link the knowledge represented in the Loom repository with information and knowledge relating to the performance optimization projects that yielded the stored knowledge. Thus, the team went with an interim solution, partially incorporating the Loom knowledge map and cases into a Lotus Notes/Domino database of project-related knowledge, to provide structure for technical lessons learned on each project. Figure 2 is a screenshot of the ported system. The immediate benefits of this included

able for knowledge verification and integrity checking.

In doing this work, we detected two weaknesses in current knowledge-engineering techniques and technology. First, as we noted, PC-PACK and other textual mark-up systems do not cope adequately with concepts defined by several nonadjacent words. Thus, we have identified the need for a more flexible tool. Second, it is very difficult to integrate expressive reasoning tools such as Loom with intranet knowledge management environments such as Lotus Notes/Domino. It seems reasonable to conclude, therefore, that while knowledge-engineering processes are ready to bring significant benefits to knowledge management projects, the knowledge-engineering toolbox needs some improvement.

Acknowledgments

The case study described here was supported by Teaching Company Scheme funding from the UK Department of Trade and Industry and the Engineering and Physical Sciences Research Council. The Lotus Notes version of the Drilling Knowledge Store was constructed largely by John Lawton and David Bowden, Transition Associates, who also proposed and developed the OASIS University. The authors thank Tom Russ of ISI/University of Southern California for advice and technical assistance regarding Loom and Ontosaurus. This article is published with the kind consent of the Hughes Christensen Company.

References

1. *Harvard Business Review on Knowledge Management*, Harvard Business School Press, Cambridge, Mass., 1998.
2. J. Liebowitz and L. Wilcox, *Knowledge Management and Its Integrative Elements*, CRC Press, Boca Raton, Fla., 1997.
3. G. Schreiber et al., *Knowledge Engineering and Management*, MIT Press, Cambridge, Mass., 2000.
4. N. Milton et al., "Towards a Knowledge Technology for Knowledge Management," *Int'l J. Human-Computer Studies*, vol. 51, no. 3, 1999, pp. 615–641.
5. E. Motta, *Reusable Components for Knowledge Modeling*, IOS Press, Amsterdam, 1999.
6. S. Russell and P. Norvig, *Artificial Intelligence*, Prentice-Hall, Old Tappan, N.J., 1995.
7. J. Domingue and E. Motta, "Planet-Onto: From News Publishing to Integrated Knowledge Management Support," *IEEE Intelligent Systems*, May/June, 2000, pp. 26–32.
8. D.A. Curry, A.V. Singelstad, and D. Bowden, "Drilling Performance Guidelines—A Tool for Sharing Drilling-Related Knowledge and Experience," SPE/IADC paper no. 52804, *SPE/IADC Drilling Conf.*, Society of Petroleum Engineers, Dallas, 1999.
9. J.M. Evans, M.J. Fear, and N.C. Meany, "A New Graphical Representation for Rule Definition and Explanation in an Expert System," *Applications and Innovations in Intelligent Systems III*, Springer-Verlag, Berlin, 1995.
10. R. Kremer, "Concept Mapping Tool to Handle Multiple Formalisms," *AAAI Spring Symp. Artificial Intelligence in Knowledge Management*, AAAI Press, Menlo Park, Calif., 1997.
11. R. MacGregor, "The Evolving Technology of Classification-Based Knowledge Representation Systems," *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, Morgan Kaufmann, San Francisco, 1991, pp. 385–400.
12. J-L. Ermine, "Knowledge Management in the Commissariat a l'Energie Atomique," *PAKeM98—Practical Application of Knowledge Management*, Practical Applications Co., London, 1998.
13. W.J. Clancey, "Model Construction Operators," *Artificial Intelligence*, vol. 53, 1992, pp. 1–115.
14. R.J. Brachman et al., "Living with Classic: When and How to Use a KL-ONE-like Language," *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, Morgan Kaufmann, San Francisco, 1991, pp 401–456.
15. S. White, *Enhancing Knowledge Acquisition with Constraint Technology*, doctoral dissertation, Dept. Computing Science, University of Aberdeen, Aberdeen, UK, 2000.

Docket No. **9001RF-34875**

DECLARATION FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe that I am an original, first, and joint inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled

CASE-BASED DRILLING KNOWLEDGE MANAGEMENT SYSTEM

further identified by attorney docket no. **9001RF-34875.**

This application claims the benefit of U.S. Provisional Application Serial No. **60/140,119**, filed **18 JUNE 1999**, entitled **Case-Based Drilling Knowledge Management System.**

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the Office all information known to my person to be material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, Sec. 1.56(a).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

I hereby appoint **Melvin A. Hunn, Reg. No. 32,574** and **Kenneth C. Hill, Reg. No. 29,650** to prosecute this application and to transact all business in the U.S. Patent and Trademark Office in connection therewith.

Exhibit D

Please send all correspondence to:

Melvin A. Hunn
Registration No. 32,574
HILL & HUNN LLP
201 Main Street, Suite 1440
Fort Worth, Texas 76102-3105
817/332.2113

Inventor's Signature: ________________

Full Name of Inventor: **Alan Nicholas Flett**

Date of Signature: 21/8/2000

Residence and Post Office Address:
8 Mill Lade Wynd, Danestone, Aberdeen AB22 8QN, Scotland

Citizenship: United Kingdom

Inventor's Signature: ________________

Full Name of Inventor: **Derek H. Sleeman**

Date of Signature: 3 Sept 2000

Residence and Post Office Address:
10 The Chanonry, Aberdeen AB24 1RN, Scotland

Citizenship: United Kingdom

Inventor's Signature: ________________

Full Name of Inventor: **Alun D. Preece**

Date of Signature: 4/9/2000

Residence and Post Office Address:
10 Fare Park Drive, Westhill, Aberdeenshire AB32 6WE, Scotland

Citizenship: United Kingdom

PTO/SB/96 (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: Alan Nicholas Flett, et al.

Application No./Patent No.: 09/595,550 Filed/Issue Date: June 16, 2000

Titled: Case Based Drilling Knowledge Management System

Baker Hughes Incorporated, a Corporation
(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.

states that it is:

1. [x] the assignee of the entire right, title, and interest in;

2. [ ] an assignee of less than the entire right, title, and interest in
(The extent (by percentage) of its ownership interest is ______ %); or

3. [ ] the assignee of an undivided interest in the entirety of (a complete assignment from one of the joint inventors was made)

the patent application/patent identified above, by virtue of either:

A. [x] An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel 011288, Frame 0676, or for which a copy therefore is attached.

OR

B. [ ] A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: ______ To: ______
The document was recorded in the United States Patent and Trademark Office at
Reel ______, Frame ______, or for which a copy thereof is attached.

2. From: ______ To: ______
The document was recorded in the United States Patent and Trademark Office at
Reel ______, Frame ______, or for which a copy thereof is attached.

3. From: ______ To: ______
The document was recorded in the United States Patent and Trademark Office at
Reel ______, Frame ______, or for which a copy thereof is attached.

[ ] Additional documents in the chain of title are listed on a supplemental sheet(s).

[ ] As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (*i.e.*, a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Sandra E. Alford
Signature

______
Date

Sandra E. Alford
Printed or Typed Name

Corporate Secretary
Title

HCC Approval:
Finance/Acct ______
Marketing/Ops ______
Engr/Tech ______
Legal BSD
VP ______

LHH4-15390-US

PTO/SB/81 (01-09)
Approved for use through 11/30/2011. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

POWER OF ATTORNEY OR REVOCATION OF POWER OF ATTORNEY WITH A NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS

Application Number	09/595,550
Filing Date	June 16, 2000
First Named Inventor	Alan Nicholas Flett, et al.
Title	Case Based Drilling Knowledge Manag
Art Unit	2123
Examiner Name	Thomas H. Stevens
Attorney Docket Number	TA-00418.0

I hereby revoke all previous powers of attorney given in the above-identified application.

☐ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith: 35979

OR

☐ I hereby appoint Practitioner(s) named below as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

Practitioner(s) Name	Registration Number

Please recognize or change the correspondence address for the above-identified application to:

☒ The address associated with the above-mentioned Customer Number.

OR

☐ The address associated with Customer Number:

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the:

☐ Applicant/Inventor.

OR

☒ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) (Form PTO/SB/96) submitted herewith or filed on ____________.

SIGNATURE of Applicant or Assignee of Record

Signature	Sandra E. Alford	Date	
Name	Sandra E. Alford	Telephone	713-439-8673
Title and Company	Corporate Secretary		

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

☐ *Total of ________ forms are submitted.

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

HCC Approval:
Finance/Acct ____
Marketing/Ops ____
Engr/Tech ____
Legal DEO ____
VP ____

HH44-15390-US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **ALAN NICHOLAS FLEET ET AL**	§	Attorney Docket No.: **TA-00418.0**
	§	
Serial No.: **09/595,550**	§	Examiner: **STEVENS, THOMAS H..**
	§	
Filed: **JUNE 16, 2000**	§	Art Unit: **2123**
	§	
For: **CASE BASED DRILLING KNOWLEDGE MANAGEMENT SYSTEM**	§	Confirmation No.: **1122**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO SUPPLEMENTAL NOTICE OF ALLOWABILITY DATED NOVEMBER 18, 2005

Dear Sir/Madam:

Applicant files this timely response to the Supplemental Notice of Allowability mailed November 18, 2005, together with a petition to revive the above titled application and necessary fees. If any additional fees are necessary for continued prosecution of this Application, the commissioner is hereby authorized to charge those fees to **Bracewell & Giuliani LLP's Deposit Account Number 50-0259 (9001RF.045342).**

Abstract is attached.

Substitute Specification is attached.

Remarks begins on page 2 of this paper.

REMARKS ONLY

The Office has objected to the specification and noted the lack of an abstract in the specification as filed. Applicant submits herewith a substitute specification correcting minor grammatical errors and adding a description of the drawings. Applicant also submits an abstract. Accordingly, it is respectfully requested that the objections to the specification be withdrawn.

As for the lack of an oath or declaration in this matter, Applicant respectfully submits that this objection is in error. An oath or declaration in this matter was transmitted to the USPTO on November 14, 2000, as shown in the file record. The file record of the declaration does not contain a signature page, however. Applicant discovered the signature page for this declaration was recorded with the inventors' assignment of the above titled application on November 17, 2000, at Reel/Frame 011288/0676, the complete oath being attached hereto as Exhibit A. In this regard, Applicant also notes that the Office failed to change the name Alan Fleet to Alan Flett per the declaration and assignment papers.

Date: 8/1/11

Respectfully submitted,

Ashley L. Kirk
Registration No. 51,261
BRACEWELL & GIULIANI LLP
P.O. Box 61389
Houston, Texas 77208-1389
(713) 221-1544 direct phone
(713) 437-5381 direct facsimile
Email: ashley.kirk@bgllp.com
Attorney for Applicant

ABSTRACT

A method has been designed for storing drilling knowledge and experience in a highly structured fashion that permits the user to identify drilling cases that meet user-specified criteria and to retrieve the knowledge and experience relating to those cases. In this way the user is able to retrieve the knowledge and experience learned in cases that are analogous to one or more current cases they are studying.

SUBSTITUTE SPECIFICATION

NON-PROVISIONAL PATENT APPLICATION

BACKGROUND OF THE INVENTION

A method has been designed for storing drilling knowledge and experience in a highly structured fashion that permits the user to identify drilling cases that meet user-specified criteria and to retrieve the knowledge and experience relating to those cases. In this way the user is able to retrieve the knowledge and experience learned in cases that are analogous to one or more current cases they are studying.

SUMMARY OF THE INVENTION

The fundamental functionality of the system is to represent drilling experiences in a highly structured fashion, so that the system can then be interrogated by querying for analogous cases. In this way, it supports certain decisions the engineer may make through the application of captured and stored organizational experience. The system's preferred technology for implementing this is a logic-intensive computer system (implemented using a Description Logic called LOOM), which allows for the logical representative of concepts and relationships commonly conceptualized in the drilling domain. These are then organized into a subsumption hierarchy automatically by LOOM; such a set of defined concepts and relationship is called an ontology in the literature. More complex concepts can be built/described using the more base/primitive concepts and relationships, such as a schema for the describing of bit run expectations elicited from field engineers including their decisions. The system can also be used to constrain what is expressible by the field engineer in terms of the case's description, thereby limiting the bounds of his knowledge, and effectively putting extreme best practice limits, on,

say, his drill bit selection in certain hardnesses of rocks. The invention claimed is the utilization of a set of concepts and relationships to represent the drilling related knowledge and experience.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts one example of drilling operations conducted in accordance with the present invention;

Figure 2 is a block diagram representative of the preferred embodiment of the present invention; and

Figure 3 is a general purpose data processing system according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

OVERVIEW OF DRILLING OPERATIONS: **Figure 1** depicts one example of drilling operations conducted in accordance with the present invention.

As is shown, a conventional rig **3** includes a derrick **5**, derrick floor **7**, draw works **9**, hook **11**, swivel **13**, kelly joint **15**, and rotary table **17**. A drillstring **19** which includes drill pipe section **21** and drill collar section **23** extends downward from rig **3** into wellbore **1**. Drill collar section **23** preferably includes a number of tubular drill collar members which connect together, including a measurement-while-drilling logging subassembly and cooperating mud pulse telemetry data transmission subassembly, which are collectively referred to hereinafter as "measurement and communication system **25**".

During drilling operations, drilling fluid is circulated from mud pit **27** through mud pump **29**, through a desurger **31**, and through mud supply line **33** into swivel **13**. The drilling mud flows through the kelly joint and an axial central bore in the drillstring. Eventually, it exits through jets which are located in downhole drill bit **26** which is connected to the lowermost portion of

measurement and communication system **25**. The drilling mud flows back up through the annular space between the outer surface of the drillstring and the inner surface of wellbore **1**, to be circulated to the surface where it is returned to mud pit **27** through mud return line **35**. A shaker screen (which is not shown) separates formation cuttings from the drilling mud before it returns to mud pit **27**.

Preferably, measurement and communication system **25** utilizes a mud pulse telemetry technique to communicate data from a downhole location to the surface while drilling operations take place. To receive data at the surface, transducer **37** is provided in communication with mud supply line **33**. This transducer generates electrical signals in response to drilling mud pressure variations. These electrical signals are transmitted by a surface conductor **39** to a surface electronic processing system **41**, which is preferably a data processing system with a central processing unit for executing program instructions, and for responding to user commands entered through either a keyboard or a graphical pointing device.

The mud pulse telemetry system is provided for communicating data to the surface concerning numerous downhole conditions sensed by well logging transducers or measurement systems that are ordinarily located within measurement and communication system **25**. Mud pulses that define the data propagated to the surface are produced by equipment which is located within measurement and communication system **25**. Such equipment typically comprises a pressure pulse generator operating under control of electronics contained in an instrument housing to allow drilling mud to vent through an orifice extending through the drill collar wall. Each time the pressure pulse generator causes such venting, a negative pressure pulse is transmitted which can be received by surface transducer **37**. An alternative conventional arrangement generates and transmits positive pressure pulses. As is conventional, the circulating mud provides a source of

energy for a turbine-driven generator subassembly which is located within measurement and communication system **25**. The turbine-driven generator generates electrical power for the pressure pulse generator and for various circuits including those circuits which form the operational components of the measurement-while-drilling tools. As an alternative or supplemental source of electrical power, batteries may be provided, particularly as a back-up for the turbine-driven generator.

OVERVIEW OF THE INVENTION

Figure 2 is a block diagram representative of the preferred embodiment of the present invention. As is shown, a drilling situation **101** is presented to user **103**. User **103** must make a decision concerning the drilling situation. The decision may include determining what types of downhole equipment to utilize, such as a selection of rock bits for particular types of drilling situations. User **103** interacts through user interface **107** with database **111**. User interface includes a means for receiving a search request from user **103** and a means for providing an output to user **103** in a human-readable form. In accordance with the preferred embodiment of the present invention, the user generates a query which is defined by user-specified criteria **109** which is passed from user interface **107** to database **111**. Database **111** includes structured data which represents captured and stored organizational experience. For example, the structured data may include drilling knowledge **115** and/or drilling experience **117**. The user-specified criteria is utilized to query database **111** in a manner which generates as an output the relevant or analogous knowledge and experience resident or present in database **111**. This is passed through user interface **107** to user **103**. User **103** then may utilize the knowledge to make drilling decision **105**.

The user interface **107** and database **111** of **Figure 2** are preferably constructed utilizing executable program instructions. Preferably, the program instructions are executed by a general purpose data processing system, such as that depicted in **Figure 3**.

With reference now to the figures and in particular with reference to **Figure 3**, there is depicted a pictorial representative of data processing system **41** which may be programmed in accordance with the present invention. As may be seen, data processing system **41** includes processor **12** which preferably includes a graphics processor, memory device and central processor (not shown). Coupled to processor **12** is video display **14** which may be implemented utilizing either a color or monochromatic monitor, in a manner well known in the art. Also coupled to processor **12** is keyboard **16**. Keyboard **16** preferably comprises a standard computer keyboard which is coupled to the processor by means of cable **18**.

Also coupled to processor **12** is a graphical pointing device, such as mouse **20**. Mouse **20** is coupled to processor **12**, in a manner well known in the art, via cable **22**. As is shown, mouse **20** may include left button **24**, and right button **26**, each of which may be depressed, or "clicked", to provide command and control signals to data processing system **41**. While the disclosed embodiment of the present invention utilizes a mouse, those skilled in the art will appreciate that any graphical pointing device such as a light pen or touch sensitive screen may be utilized to implement the method and apparatus of the present invention. Upon reference to the foregoing, those skilled in the art will appreciate that data processing system **41** may be implemented utilizing a so-called personal computer.

DESCRIPTION OF LOOM

LOOM is a research project in the Artificial Intelligence Research Group at the University of Southern California's Information Sciences Institute. The goal of the project is to develop and field advanced tools for knowledge representation and reasoning in Artificial Intelligence. LOOM software is the intellectual property of the University of Southern California.

The LOOM software is a language and environment for constructing intelligent applications. At the heart of LOOM is a knowledge representation system used to provide deductive support for a declarative knowledge portion of the LOOM language. Declarative knowledge in LOOM consists of definitions, rules, facts, and default rules. A deductive engine called a classifier utilizes forward-chaining, semantic unification, and object-oriented truth maintenance technologies to compile the declarative knowledge into a network designed to efficiently support on-line deductive query processing.

The LOOM system implements a logic-based pattern matcher for driving a production rule facility, and a pattern-directed method dispatching facility for supporting the definition of object-oriented methods. There is a high degree of integration between LOOM's declarative and procedural components. This permits programmers to utilize logic programming, production rule, and object-oriented programming paradigms in a single application. The LOOM software can also be used as a deductive layer that overlays an ordinary CLOS network. In this mode, users can obtain many of the benefits of using LOOM without impacting the function or performance of their CLOS based applications.

LOOM has been distributed to more than 80 universities and corporations, and is being used in numerous DARPA-sponsored projects in planning, software engineering, and intelligent integration of information.

Licensing of LOOM:

The LOOM software is the intellectual property of the University of Southern California. It is not in the public domain; however, the University of Southern California grants permission to use this software for non-commercial purposes without a fee. If an application is covered by the terms of this fee-free license, it is not necessary to execute a written license agreement. More information about LOOM itself, its availability, and commercial licenses may be obtained from U.S.C. Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, California 90292-6695.

System Requirements:

The LOOM software requires a Common Lisp compiler to function properly. LOOM can be expected to function properly in an ANSI-compliant Common Lisp. LOOM has been tested using the following Common Lisp compilers and platforms: Macintosh Common Lisp version 2.1-4.2; Franz Allegro Common Lisp; Unix (Sparc) versions 4.1-4.3, 5.0; Windows version 4xxx, 5.x (NT); Harlequin LispWorks; Unix (Sparc) version 3.0.2; Windows version 4.0.1 (NT); and Lucid Common Lisp for Unix version 4.1. Although other Lisp systems may work as well, they are not known to have been tested. Newer versions of the lisp systems from these vendors should work. For example, user reports exist which indicate that LOOM works with CMU Common Lisp version 18b.

Source Code:

The LOOM software is distributed as source files which must be compiled using a Common Lisp compiler. Several versions of LOOM are available. Version 4.0 is the current release. The LOOM software is available in several formats,

The LOOM 4.0 Source includes installation instructions, and release notes in at least the following files: Unix tar file (3.9 MB), Gzipped Unix tar file (866 kB), and Macintosh binhex file (1.4 MB).

The LOOM 3.0 Source includes installation instructions, and release notes in at least the following files: Unix tar file (3.5 MB), Gzipped Unix tar file (831 kB), and Macintosh binhex file (1.44 MB).

The LOOM 2.1 Source includes installation instructions, and release notes in at least the following files, Unix tar file (3.3 MB) and Gzipped Unix tar file (780 kB).

Documentation, Descriptive Papers, and Manuals:

1. *LOOM Reference Manual* for LOOM version 2.0, Dave Brill, December 1993. The full paper version is available on-line and in two formats as a postscript file. Note that since the *Reference Manual* is quite old, the release notes are an important additional source of documentation.

2. Preliminary *LOOM Function Card* for LOOM version 4.0. A quick summary of LOOM functions, December 1998.

3. *LOOM Function Card* for LOOM version 2.0. A quick summary of LOOM functions, December 1993. The full paper is available in postscript.

4. *Tutorial* for LOOM version 2.1. May 1995. The full paper is available in postscript.

5. LOOM *User's Guide* for LOOM version 1.4. ISX Corporation, August 1991. The full paper is available in postscript.

Additional Information:

From the Intelligent Systems Division of ISI/UCS, Los Angeles, CA, USA.

1. Useful Loom reference material:

The Intelligent Systems Division of Southern California hosts a website (isi.edu/isd) which provides LOOM reference material. Such materials include: Loom Users Guide (more structured description of language), Loom Tutorial (introductory examples and basic concepts), Loom Reference Manual (reference manual of all functions, macros, constructs, grammar, etc.).

2. Ontosaurus: From the Intelligent Systems Division of ISI/UCS, CA, USA.

THE KNOWLEDGE BASE

There are two main parts to the knowledge base: a conceptual model part, and a database part. This is analogous to a database with its schema and data parts. The conceptual part of the knowledge base is defined using concepts. There are binary concepts (otherwise known as roles or relations) and unary concepts (otherwise known as concepts or classes). A introduction to these ideas in terms of Loom can be found in the Loom reference material.

The existential database part maybe easier to edit. To this end a brief summary is given of what the new instances should look like when being entered. The constructors most easily used in the database part of the knowledge base are "tell" and "about". Tell is used to assert propositions and facts about the world or domain. About references the instance those propositions refer to.

Domain Specific Conceptual Modelling and Case Capture: This is in general how a case is to be entered.

<u>CASE:</u> The instance below is a case instance. It has one formation sequence name and zero or more decisions and/or observations.

```
(tell (:about Case-Name
        CASE
        (formation-sequence Formation-Sequence-Name)
        (decision Decision-Name)
        (observation Observation-Name)))
```

The instance below is that of a formation sequence instance. It was one or more formation names.

```
(tell (:about Formation-Sequence-Name
        FORMATION_SEQUENCE
        (formation Formation-Name)))
```

The instance below is that of a formation. It has one or more lithologies. It can also have start and stop depths and an interval length.

```
(tell (:about Formation-Name
        FORMATION
        (lithology Lithology-Name)
        (formation-interval-length-ft/m NUMBER)))
```

The instance below is a lithology. It has one or more hardness definitions. The one below has a lithology hardness percent number meaning that there is number percent of the lithology with the hardness specified by the hardness instance in the ternary relation. Using the interval length from the formation this lithology is in, the lithology hardness amount in feet or meters can be calculated.

```
(tell (:about Lithology-Name
```

```
LITHOLOGY
(lithology-hardness-percent Generic-Hardness-Instance-Name NUMBER)
(lithology-type-percent NUMBER)))
```

The lithology definitions below has the lithology hardness amount entered directly.

```
(tell (:about Lithology-Name
    LITHOLOGY
    (lithology-hardness-amount-ft/m Generic-Hardness-Instance-Name NUMBER)
    (lithology-type-amount-ft/m NUMBER)))
```

Below is a decision instance. It has goal concepts and issue concepts. It also has an action instance, an author instance, and a reasoning string. (There could be other roles too.)

```
(tell (:about Decision-Name
    DECISION
    (goal GOAL)
    (issue ISSUE)
    (action Action-Name)
    (author Author-Name)
    (authors-reasoning STRING)))
```

Below is a drill bit action instance. It has a drill bit role which is to be filled by a drill bit concept.

```
(tell (:about Action-Name
    ACTION
    (drill-bit DRILL_BIT)))
```

The present invention utilizes an ontological system which employs the LOOM modeling code which is available over the internet from the Information Sciences Institute. LOOM 3.0 is the version that is currently available over the internet. The following is a description of the ontological system, and includes representative code for modeling drilling information.

THE ONTOLOGICAL SYSTEM: The system's main ontological definitions will now be outlined with both a plain English style explanation of the intended meaning and with a copy of the Loom modeling code to reference.

KNOWLEDGE MANAGEMENT CONCEPTS

This is one of the main ontological "top level concepts". Top level concepts are those that are of type thing, which is a concept defined in Loom's theory BUILT-IN-THEORY. Any user-defined concept will be of type THING.

(defconcept KNOWLEDGE_MANAGEMENT_CONCEPT)

Knowledge management concepts can be broken down into three broad categories including I CASES, II ENVIRONMENT, III DOWNHOLE EQUIPMENT. The following is a description of each of these three categories. The types of information which is modeled in the CASES category includes date, location, decisions, issues, actions, goals, author, author's reasoning, company's reasoning, spin, observation, observation text, observation category, and observation formation. The information which is structured in the environment category includes drilling fluid concepts, rock concepts, formation sequences, single formation sequences, multiple formation sequences, formations, single ontology formations, multiple ontology formations, lithologies, and hardnesses. The types of information organized in the downhole equipment category includes bottomhole assemblies, bottomhole assembly components, and drill bits.

I. CASES

A case is (usually) a bit run case. That is, if some one experienced some bit run worthy of broadcast, then the formation sequence drilled should be represented, as well as the decisions taken on how to drill that formation sequence. That is, there are several decisions which must also be represented. A decision can be a decision about the chosen drill bit, the mud, the BHA,

the flow rate, and so on. The case need not be an actual drill bit run if the person entering the case feels that he has some highly structured experience or knowledge he wishes to share. A case also has a date on which it was captured and a location which it is supposed that the case is applicable to. Representative code for case identification follows.

```
(defconcept CASE
        :is-primitive
        (:and (:exactly 1 formation-sequence)
        (:all decision DECISION)
        (:all observation OBSERVATION)))
```

(1) <u>Date</u>: The date is the date on which the bit run or generic knowledge was 'told' to the KB. It consists of at most 1 day, month, and/or year. Representative code follows.

```
(defconcept DATE :is-primitive
        (:and (:at-most 1 day)
                (:at-most 1 month)
                (:at-most 1 year)))
```

(2) <u>Location</u>: The location is the geographical position at which the bit run or knowledge is believed to be applicable to or came from. Representative code follows.

```
(defconcept LOCATION)
```

(3) <u>Decisions</u>: A decision here is presupposed to have several different dimensions. These include: issues, actions, goals, an author, a spin, and a reasoning. These dimensions, it is felt, provide a good balance between structure and text. The structure is to enable the formal representative and therefore search power required, and the textual for the more free flowing and readable information.

These decisions are parameter led in that you should only really have a decision object if you came to some decision about a parameter which is possible to modify in the real world and which was indeed modified with reference to some consideration (issue and goal). A parameter can be a drill bit, the flow rate, the BHA used, etc. Representative code follows.

```
(defconcept DECISION
        :is-primitive
        (:and (:exactly 1 action)
        (:at-most 10 issue)
        (:at-most 10 goal)
        (:at-most 1 authors-reasoning)
        (:at-most 1 companys-reasoning)
        (:at-most 1 author)
        (:at-most 1 spin)))
```

(4) <u>Issues</u>: An issue is some observation or issue that the engineer was aware of when making his decision. The issues in the current version reflect quite strongly the Best Practice Drilling database, and the link roles shown below are intended to reflect this. These could be filled with links to other media, indeed to the Notes database itself through URLs. Representative code follows.

```
(defconcept ISSUE
        :is-primitive
        (:and KNOWLEDGE_MANAGEMENT_CONCEPT
                (:at-most 1 symptoms-and-diagnosis-link)
                (:at-most 1 description-link)
                (:at-most 1 parameters-link)
                (:at-most 1 diagnostic-information-link)
                (:at-most 1 planning-actions-link)
                (:at-most 1 operating-practices-link)
                (:at-most 1 examples-link)))
```

(5) Actions: An action is the real-world consequent the engineer performed as part of his decision. Representative code follows.

```
(defconcept ACTION
    :is-primitive
    (:and KNOWLEDGE_MANAGEMENT_CONCEPT
    (:at-most 1 categorical-outcome)
    (:at-most 1 textual-outcome)))
```

(6) Goals: A goal is an objective which the engineer was aiming to achieve with that decision in particular. Representative code follows.

```
(defconcept GOAL
    ""
    :is-primitive
    (:and KNOWLEDGE_MANAGEMENT_CONCEPT))
```

(7) Author: The author is the name of the person telling the decision to the KB. This is part of the decision and not the case as there may be a team involved, with different experts taking different decisions, or there may be multiple experts involved in the one decision. Representative code follows.

```
(defconcept AUTHOR :is-primitive
    (:and (:exactly 1 first-name)
        (:exactly 1 last-name)))
```

(8) Author's Reasoning: Reasoning is a field of free-text for explanations for example of why a certain drill bit was chosen. This is to allow the possibility of incomplete, inaccurate, and even incoherent explanations for actions being stored; after all, the main reasoning or

determinism for the action is the other structured information describing the circumstances in which the action was taken, such as the formation sequence. Representative code follows.

```
(defrelation authors-reasoning
        :domain DECISION
        :range STRING
        :characteristics (:closed-world :single-world))
```

(9) <u>Company's Reasoning</u>: This is to allow a field which expresses the company's commonly agreed upon beliefs for the decision in question. Representative code follows.

```
(defrelation companys-reasoning
        :domain DECISION
        :range STRING
        :characteristics (:closed-world :single-world))
```

(10) <u>Spin</u>: The 'spin' is the type of the knowledge captured in terms of whether it is positive or negative in its effect. That is, if a case is entered which in effect is guiding the user to not using the action used in that example, then that should be considered a type of negative knowledge. If, on the other hand, a case has as its effect an engineer retrieving a case and executing the same or similar action, then that case should be considered positive. Representative code follows.

```
(defrelation spin
        :domain DECISION
        :range SPIN
        :characteristics (:closed-world :single-world))

(defset SPIN
        :is-primitive
        (:and KNOWLEDGE_MANAGEMENT_CONCEPT
              (:one-of 'Positive 'Negative)))
```

(11) Observations: An observation is some state or behavior observed in the drilling process. The concept has the three roles of textual-observation, categorical-observation, and observation formation. The last of these relations allows the user to specify more accurately where the observation is valid. Observations should be left to the sort of knowledge exemplified by "ROP dips going through the sand stringers lower down in this bit run", etc. Representative code follows.

```
(defconcept OBSERVATION
        ""
        :is-primitive
        (:and KNOWLEDGE_MANAGEMENT_CONCEPT
                (:at-most 1 observation-formation)
                (:at-most 1 textual-observation)
                (:at-most 1 categorical-observation)))
```

(12) Observation Text: A free text field left for the user to explain himself in plain English. Representative code follows.

(13) Observation Category: This is a concept hierarchy of observations which help to narrow for the purposes of search the context, subject, and object, amongst other things, of the free text.

(14) Observation Formation: This is to allow the user to more accurately contextualize his observation.

II. ENVIRONMENT

The environment here is described in terms of conceptual rock sequences. These sequences have some ontological constraints on them. For instance, if the user wishes to specify the depth and/or length of a particular section of lithology then that section has to be represented as a formation.

The super-structure larger than that is the formation sequence, which can have one or more formations. Each formation can have one or more lithology. A lithology has no depth or length roles.

(1) Drilling Fluid Concepts: This is a concept representing the muds used in drilling oil wells. This is a top level concept in the ontology. Representative code follows.

```
(defconcept DRILLING_FLUID_CONCEPT)
```

(2) Rock Concepts: This is a top level concept representing all rock-related concepts, including lithologies, formations, and formation sequences. Representative code follows.

```
(defconcept ROCK_CONCEPT)
```

(3) Formation Sequences: A formation sequence is the constructor used to assemble all of the separate sub-sequences identified by the user when describing their well. The formation sequence will usually be the section the drill bit drilled in its run. A formation sequence has one or more constituent formations. Representative code follows.

```
(defconcept FORMATION_SEQUENCE
        :is-primitive
        (:and ROCK_CONCEPT
                (:at-least 1 formation)))
```

(4) Single Formation Formation Sequences: This is a formation sequence with only the single constituent formation. Representative code follows.

```
(defconcept SINGLE_FORMATION_FORMATION_SEQUENCE
        :is
        (:and FORMATION_SEQUENCE
        (:exactly 1 formation)))
```

(5) <u>Multiple Formation Formation Sequences</u>: This is a formation sequence with at least two constituent formations. Representative code follows.

```
(defconcept MULTIPLE_FORMATION_FORMATION_SEQUENCE
        :is
        (:and FORMATION_SEQUENCE
        (:at-Ieast 2 tormatlonjj)
```

(6) <u>Formations</u>: A formation has one or more lithologies. A formation is the conceptual modeling granularity at which the user should be representing any part of his well he feels should have represented interval lengths and depths. Even stringers, or production sands, if these, it is felt, need to have depths and/or lengths attached to their storage, would need to be modeled as 'formations'. If however, it is only stringers in a majority lithology, which have unimportant positions of depth in the over-all sequence due to uncertainty or irrelevance, then these stringers are 'lithologies' and the overall stringer and majority lithology sequence is then the 'formation'. Representative code follows.

```
(defconcept FORMATION
        :is-primitive
        (:and ROCK_CONCEPT
        (:at-Ieast 1 lithology)))
```

The relation comes-in-somewhere-after relates two formations, the first of which comes in somewhere after the other. Representative code follows.

```
(defrelation comes-in-somewhere-after
        :domain FORMATION_SEQUENCE
        :range FORMATION
        :characteristics (:multiple-valued :closed-world)
        :is (:satisfies (?formation-x ?formation-y)
                (:and
```

```
(FORMATION ?formation-x)
(FORMATION ?formation-y)
(:or (comes-in-immediately-after ?formation-x ?formation-y)
    (:exists (?formation-z)
        (:and
        (FORMATION ?formation-z)
        (comes-in-somewhere-after ?formation-x ?formation-z)
        (comes-in-somewhere-after ?formation-z ?formation-
        y)))))))
```

The relation comes-in-somewhere-before relates two formations, the first of which comes in somewhere before the other. Representative code follows.

```
(defrelation comes-in-somewhere-before
    :domain FORMATION_SEQUENCE
    :range FORMATION
    :characteristics (:closed-world :multiple-valued)
    :is (:satisfies (?formation-x ?formation-y)
        (:and
        (FORMATION ?formation-x)
        (FORMATION ?formation-y)
        (:or (comes-in-immediately-before ?formation-x ?formation-y)
            (:exists (?formation-z)
                (:and
                (FORMATION ?formation-z)
                (comes-in-somewhere-before ?formation-x ?formation-z)
                (comes-in-somewhere-before ?formation-z ?formation-
                y)))))))
```

The relation comes-in-immediately-after relates two formations, the first of which comes in immediately after the other. Representative code follows.

```
(defrelation comes-in-immediately-after
        :domain FORMATION_SEQUENCE
        :range FORMATION
        :characteristics (:closed-world :single-valued))
```

The relation comes-in-immediately-before relates two formations, the first of which comes in immediately before the other. Representative code follows.

```
(defrelation comes-in-immediately-before
        :domain FORMATION_SEQUENCE
        :range FORMATION
        :characteristics (:closed-world :single-valued))
```

(7) <u>Single Lithology Formations</u>: These are formations conceptualized to have only the single lithology type present (although it may have multiple harnesses). Representative code follows.

```
(defconcept SINGLE_LITHOLOGY_FORMATION
        :is
        (:and FORMATION
                (:exactly 1 lithology)))
```

(8) <u>Multiple Lithology Formations</u>: These are formations conceptualized to have multiple lithology types present (each of which may have multiple harnesses). Representative code follows.

```
(defconcept MULTIPLE_LITHOLOGY_FORMATION
        :is
        (:and FORMATION
        (:at-least 2 lithology)))
```

(9) Lithologies: A lithology is the basic rock type, e.g., sand, shale, etc. Each lithology type also has an amount or a percentage amount for the formation in which it is present. As well as this, each lithology type is broken down into having a hardness, which then itself has an amount or a percentage amount for the lithology type. In this way, complex conceptual descriptions can be built up of the well's statistical hardness profile. It can be envisaged that a tool useful in the construction of such conceptual description could be useful; as different levels of accuracy are obtainable, from the very coarse, to the very fine, grained and, obviously, the more fine grained, the more work is required to enter the definitions. Representative code follows.

```
(defconcept LITHOLOGY
        :is-primitive
        (:and
        ROCK_CONCEPT))
```

(10) Hardnesses: There are various hardness concepts, each with their meaning specified by their UCS ranges. At the top level there is HARDNESS. Below that there are six second tier sub-categories of hardness starting at OkPSI, and increasing in steps of 5kPSI. There are also finer grained hardness sub-categories below each second tier category. These start at 0 kPSI and increase in steps of 2.5kPSI. Each of the six second tier hardness categories has a LOWER_ and an UPPER_ third tier sub-category. This gives twelve third-tier hardness categories.

An interesting note to be made about harnesses is the following. If there is known to be 60% hard shale and 40% firm shale then that shale can be modeled with those stated percentages. If, however, it is only known that the shale has firm and hard harnesses in it, then a hardness FIRM_TO_HARD could be used as its hardness category (of which there is obviously 100% of). This modelling trick is useful when the exact percentage breakdown of the various constituent harnesses is unavailable. Representative code follows.

```
(defconcept HARDNESS
        :is-primitive
        (:and ROCK_CONCEPT
                (:at-most 1 ucs)))
```

Due to the fact that a lithology may have one type (e.g., it is shale), they can have more than one hardness (e.g., that shale may have 100m of very soft rock with 300m soft rock also). Representative code follows.

```
(defrelation hardness
        :domain LITHOLOGY
        :range HARDNESS
        :characteristics (:closed-world :multiple-valued))
```

The following relation calculates the amount of lithology in feet of a certain hardness. It relates the amount, the hardness and the lithology in a ternary relation.

```
(defrelation lithology-hardness-amount-ft
        :arity 3
        :domains (LITHOLOGY HARDNESS)
        :range NUMBER
        :characteristics (:closed-world :single-valued)
        :is (:satisfies (?lithology ?hardness ?hardness-amount-ft)
                (:exists (?percent ?amount)
                        (:and (lithology-hardness-percent ?lithology ?hardness ?percent)
                                (lithology-type-amount-ft ?lithology ?amount)
                                (* (/ ?percent 100.0)?amount ?hardness-amount-ft)))))
```

The following relation calculates the amount of lithology in meters of a certain hardness. It relates the amount, the hardness and the lithology in a ternary relation.

```
(defrelation lithology-hardness-amount-m
```

```
        :arity3
        :domains(LITHOLOGY HARDNESS)
        :rangeNUMBER
        :characteristics (:closed-world :single-valued)
        :is (:satisfies (?lithology ?hardness ?hardness-amount-m)
                (:exists (?percent ?amount)
                    (:and (lithology-hardness-percent ?lithology ?hardness ?percent)
                        (lithology-type-amount-m ?lithology ?amount)
                        (* (/ ?percent 100.0)?amount ?hardness-amount-m)))))
```

The following relation calculates the amount of lithology in a formation in feet.

```
    (defrelation lithology-type-amount-ft
        :domain LITHOLOGY
        :rangeNUMBER
        :characteristics (:closed-world :single-valued)
        :is (:satisfies (?lithology ?amount-ft)
                (:exists (?percent ?formation ?length-ft)
                    (:and (lithology ?formation ?lithology)
                        (formation-interval-length-ft ?formation ?length-ft)
                        (lithology-type-percent ?lithology ?percent)
                        (* (/ ?percent 100.0)?length-ft ?amount-ft)))))
```

The following relation calculates the amount of lithology in a formation in meters.

```
    (defrelation lithology-type-amount-m
        :domain LITHOLOGY
        :rangeNUMBER
        :characteristics (:closed-world :single-valued)
        :is (:satisfies (?lithology ?amount-m)
                (:exists (?percent ?formation ?length-m)
                    (:and (lithology ?formation ?lithology)
                        (formation-interval-length-m ?formation ?length-m)
```

```
(lithology-type-percent ?lithology ?percent)
(* (/ ?percent 100.0) ?length-m ?amount-m)))))
```

The following concept is an example of one of the second tier hardness concepts. It will have two subconcepts, upper soft and lower soft, beneath it. It is defined in terms of its ucs range. It is also defined in that it is not primitive, and as such will recognize any hardness instance with its ucs range.

```
(defconcept SOFT
    :is
    (:andHARDNESS
    (>= ucs 5000)
    (<ucs 10000)))
```

III. DOWN-HOLE EQUIPMENT

A down-hole tool is here defined to be anything that goes down hole and is not part of a recycling system (such as the fluid). So, a down-hole tool is a drill bit, a BHA component, and so on. Down-hole equipment concept is a top level concept.

```
(defconcept DOWN-HOLE_EQUIPMENT_CONCEPT)
```

(1) <u>Bottom Hole Assemblies</u>: Bottom hole assemblies have at least 1 bha component. Representative code follows.

```
(defconcept BOTTOM_HOLE_ASSEMBLY
    :is-primitive
    (:and DOWN-HOLE_EQUIPMENT_CONCEPT
    (:at-least 1 bha-component)))
```

(2) <u>BHA Components</u>: A BHA component is anything capable of being added to the BHA. So, motors, bits, MWD, VSS, under-reamers, etc, are all considered BHA_COMPONENTS. Representative code follows.

```
(defconcept BHA_COMPONENT
    :is-primitive
    (:and DOWN-HOLE_EQUIPMENT_CONCEPT))
```

(3) <u>Drill Bits</u>: A drill bit is a type of down-hole equipment concept with exactly 1 bit gauge. Representative code follows.

```
(defconcept DRILL_BIT :is-primitive
    (:and DOWN-HOLE_EQUIPMENT_CONCEPT
    (:exactly 1 bit-gauge))).
```

Following is a description of how one interacts with the database utilizing the LOOM artificial intelligence system. Representative code is included where necessary to explain the concepts.

HOW TO INTERACT WITH THE SYSTEM

I. Querying Instances

The functions retrieve and ask provide the interface to Loom's deductive query facility. Retrieve is used for retrieving facts (instances from a knowledge base), while ask is used to determine whether or not a proposition is true with respect to the currently stated rules and facts.

A query has one of the following forms

(retrieve ?m1 *query)*

(retrieve (?m1 ... ?mn)

(ask *query)*

The $?m_n$ are called output variables, and query is an open sentence in which the output variables appear unbound (unquantified). Qquery can be any arbitrary expression in the first order predicate calculus (FOPC). The output variables must be prefixed with the character '?'.

As a note of interest, when querying for instances, the full benefits of the subsumption relation comes free. That is, if you have an instance of a drill bit called My-STR20 which is asserted (or deduced) to be of type STR20 (a type of TRICONE), when you come to look for cases with drill bits which are of type TRICONE, My-STR20 will be recognized as such by Loom's nominal query behavior. However, if a case's drill bit has been, like it is here, represented as a concept, then if you enter the query looking for cases with a drill bit of type TRICONE, Loom will, unless forced, not check the conceptual subsumption relation. Hence, to search in a more powerful manner, the function subrelations must be used, which will force Loom to check for subrelation (including subconcept) relations.

II. Examples Queries

Beginning with basic queries, a more complex query will be built.

(1) Querying Formation Sequences: This is probably the most sophisticated form of querying that will be done of the knowledge base. The two concepts likely to be of interest are formation sequences and formations (which are components of formation sequences). This is due to the more complex ternary relations used in the modelling of the lithologies (which are components of formations). It is also due to the ambiguity that will arise when users split up (or 'conceptualize') their wells into different formations (whilst, maybe, talking about essentially the same formation sequence) using different criteria. If that happens, then the user will want to look for formation sequences that, as a whole, contain, for example, 300 meters and over of very soft to soft shale type lithology. This therefore involves summing the amounts of very soft -to soft

shales over all the formations constituent to each formation sequence. Hence the requirement for the sum and count functions below. Also, the query works over all sub-types of shale (e.g., balling shale). It also works over all sub-types of either soft or very soft (e.g., lower very soft). So, the lithology defined as lower very soft balling shale would also have its amount added (as per the query below) to the variable ?lithology-amount-ft.

There are several different flavors to querying formation sequences. You can query for an overall cumulative amount of a certain hardness of a certain lithology over all the formations of a formation sequence.

You can query for formations which have amounts of certain lithologies of certain hardness. The query below queries for cases which have a formation sequence which has as its constituents of its formation(s) greater than or equal to 1900 feet of very soft to soft (including all subtypes of soft and very soft) of shale (including all sub-types of shale). Representative code follows.

```
(retrieve ?case
        (:and
        (CASE ?case)
        (>= (sum (:collect ?lithology-amount-ft
                (:and
                (:exists (?formation-sequence ?formation ?lithology ?hardness)
                        (:and
                        (formation-sequence ?case ?formation-sequence)
                        (formation ?formation-sequence ?formation)
                        (lithology ?formation ?lithology)
                        (lithology-hardness-amount-ft ?lithology ?hardness ?lithology amount-ft)
                        (:or
                        (VERY_SOFT ?hardness)
                        (SOFT ?hardness))
```

```
(SHALE ?lithology)
))))) 
1900)))
```

2. Querying Authors, Dates, and Locations: The query below queries for cases that have Scott MacDonald as author, of 1998, and in the Gulf of Mexico area. Representative code follows.

```
(retrieve ?case
    (:and
    (CASE ?case)
    (:exists (?decision ?date ?year ?author ?location)
        (:and
        (date ?case ?date)
        (year ?date ?year)
        (= ?year 1998)
        (decision ?case ?decision)
        (author ?decision ?author)
        (= ?author Scott-MacDonald)
        (location ?case ?location)
        (= ?location GOM)))))
```

3. Querying Goals: The query below queries for cases that have a drill bit decision such that one of its goals was for good ROP with good bit cleaning. Representative code follows.

```
(retrieve ?case
    (:and
    (CASE ?case)
    (:exists (?decision)
        (decision ?case ?decision)
        (DRILL_BIT_PLANNING_DECISION ?decision)
        (goal ?decision GOOD_ROP_WITH_GOOD_BIT_CLEANING))))
```

4. Querying Actions: The query below queries for cases where the drill bit (planning) action was to use a drill bit of type steel-tooth. This query will work if the drill-bit role has been filled with a concept. The second query will work if the role drill bit has been filled by an instance of a drill bit. For the querying of other actions, goals, and issues, the first type of querying would be used, as all their respective roles have their fillers in instances (of cases) represented as concepts. Representative code follows.

```
(retrieve ?case
    (:and
    (CASE ?case)
    (:exists (?decision ?action ?bit)
        (:and
        (decision ?case ?decision)
        (DRILL_BIT_PLANNING_DECISION ?decision)
        (action ?decision ?action)
        (DRILL_BIT_PLANNING_ACTION ?action)
        (drill-bit ?action ?bit)
        (subrelations STEEL-TOOTH_DRILL_BIT ?bit)))))
```

```
(retrieve ?case
    (:and
    (CASE ?case)
    (:exists (?decision ?action ?bit)
        (:and
        (decision ?case ?decision)
        (DRILL_BIT_PLANNING_DECISION ?decision)
        (action ?decision ?action)
        (DRILL_BIT_PLANNING_ACTION ?action)
        (drill-bit ?action ?bit)
        (STEEL-TOOTH_DRILL_BIT ?bit)))))
```

5. Complex Querying: The query below is an example of how many smaller queries can be stuck together to generate more complex queries. Particular note should be taken of the many individual :exists calls with their attendant parameters lists. Representative code follows.

The complex query below queries for cases that have more than or equal to 1900 feet of soft to very soft shale (as above), which had as a goal of the drill bit decision good ROP with good bit cleaning (as above), which also were recorded in 1998 in the Gulf of Mexico by Scott MacDonald (as above), and that the drill bit chosen should be some type of steel tooth drill bit.

```
(retrieve ?case
        (:and
        (CASE ?case)
        10 (>= (sum (:collect ?lithology-amount-ft
                (:and
                (:exists (?formation-sequence ?formation ?lithology ?hardness)
                        (:and
                        (formation-sequence ?case ?formation-sequence)
                        (formation ?formation-sequence ?formation)
                        (lithology ?formation ?lithology)
                        (lithology-hardness-amount-ft ?lithology ?hardness
?lithology-amount-ft)
                        (:or
                        (VERY_SOFT ?hardness)
                        (SOFT ?hardness))
                        (SHALE ?lithology)
                        )))))
        1900)
(:exists (?decision ?date ?year ?author ?Iocation)
        (:and
        (date ?case ?date)
```

```
        (year ?date ?year)
        (= ?year 1998)
        (decision ?case ?decision)
        (author ?decision ?author)
        (= ?author Scott-MacDonald)
        (location ?case ?location)
        (= ?location GOM)))
(:exists (?decision ?issue)
        (:and
        (decision ?case ?decision)
        (issue ?decision ?issue)
        (subrelations ?issue GOOD_ROP_WITH_GOOD_BIT_CLEANING)
        (author ?decision Scott-MacDonald)))
(:exists (?decision ?action ?drill-bit)
        (decision ?case ?decision)
        (DRILL_BIT_PLANNING_DECISION ?decision)
        (action ?decision ?action)
        (DRILL_BIT_PLANNING_ACTION ?action)
        (drill-bit ?action ?drill-bit)
        (subrelations STEEL-TOOTH_DRILL_BIT ?drill-bit))))
```

NON-PROVISIONAL PATENT APPLICATION

BACKGROUND OF THE INVENTION

~~1.~~

~~**Field of the Invention:**~~

A method has been designed for storing drilling knowledge and experience in a highly structured fashion that permits the user to identify drilling cases that meet user-specified criteria and to retrieve the knowledge and experience relating to those cases. In this way the user is able to retrieve the knowledge and experience learned in cases that are analogous to one or more current cases they are studying.

~~2.~~ SUMMARY OF THE INVENTION~~:~~

The fundamental functionality of the system is to represent drilling experiences in a highly structured fashion, so that the system can then be interrogated by querying for analogous cases. In this way, it supports certain decisions the engineer may make through the application of captured and stored organizational experience. The system's preferred technology for implementing this is a logic-intensive computer system (implemented using a Description Logic called LOOM), which allows for the logical representative of concepts and relationships commonly conceptualized in the drilling domain. These are then organized into a subsumption hierarchy automatically by LOOM; such a set of defined concepts and relationship is called an ontology in the literature. More complex concepts can be built/described using the more base/primitive concepts and relationships, such as a schema for the describing of bit run expectations elicited from field engineers including their decisions. The system can also be used to constrain what is expressible by the field engineer in terms of the case's description, thereby

limiting the bounds of his knowledge, and effectively putting extreme best practice limits, on, say, his drill bit selection in certain hardnesses of rocks. The invention claimed is the utilization of a set of concepts and relationships to represent the drilling related knowledge and experience.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts one example of drilling operations conducted in accordance with the present invention;

Figure 2 is a block diagram representative of the preferred embodiment of the present invention; and

Figure 3 is a general purpose data processing system according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

OVERVIEW OF DRILLING OPERATIONS: **Figure 1** depicts one example of drilling operations conducted in accordance with the present invention.

As is shown, a conventional rig **3** includes a derrick **5**, derrick floor **7**, draw works **9**, hook **11**, swivel **13**, kelly joint **15**, and rotary table **17**. A drillstring **19** which includes drill pipe section **21** and drill collar section **23** extends downward from rig **3** into wellbore **1**. Drill collar section **23** preferably includes a number of tubular drill collar members which connect together, including a measurement-while-drilling logging subassembly and cooperating mud pulse telemetry data transmission subassembly, which are collectively referred to hereinafter as "measurement and communication system **25**".

During drilling operations, drilling fluid is circulated from mud pit **27** through mud pump **29**, through a desurger **31**, and through mud supply line **33** into swivel **13**. The drilling mud flows through the kelly joint and an axial central bore in the drillstring. Eventually, it exits through jets

which are located in downhole drill bit **26** which is connected to the lowermost portion of measurement and communication system **25**. The drilling mud flows back up through the annular space between the outer surface of the drillstring and the inner surface of wellbore **1**, to be circulated to the surface where it is returned to mud pit **27** through mud return line **35**. A shaker screen (which is not shown) separates formation cuttings from the drilling mud before it returns to mud pit **27**.

Preferably, measurement and communication system **25** utilizes a mud pulse telemetry technique to communicate data from a downhole location to the surface while drilling operations take place. To receive data at the surface, transducer **37** is provided in communication with mud supply line **33**. This transducer generates electrical signals in response to drilling mud pressure variations. These electrical signals are transmitted by a surface conductor **39** to a surface electronic processing system **41**, which is preferably a data processing system with a central processing unit for executing program instructions, and for responding to user commands entered through either a keyboard or a graphical pointing device.

The mud pulse telemetry system is provided for communicating data to the surface concerning numerous downhole conditions sensed by well logging transducers or measurement systems that are ordinarily located within measurement and communication system **25**. Mud pulses that define the data propagated to the surface are produced by equipment which is located within measurement and communication system **25**. Such equipment typically comprises a pressure pulse generator operating under control of electronics contained in an instrument housing to allow drilling mud to vent through an orifice extending through the drill collar wall. Each time the pressure pulse generator causes such venting, a negative pressure pulse is transmitted which can be received by surface transducer **37**. An alternative conventional arrangement generates and

transmits positive pressure pulses. As is conventional, the circulating mud provides a source of energy for a turbine-driven generator subassembly which is located within measurement and communication system **25**. The turbine-driven generator generates electrical power for the pressure pulse generator and for various circuits including those circuits which form the operational components of the measurement-while-drilling tools. As an alternative or supplemental source of electrical power, batteries may be provided, particularly as a back--up for the turbine-driven generator.

OVERVIEW OF THE INVENTION

Figure 2 is a block diagram representative of the preferred embodiment of the present invention. As is shown, a drilling situation **101** is presented to user **103**. User **103** must ~~20~~ make a decision concerning the drilling situation. The decision may include determining what types of downhole equipment to utilize, such as a selection of rock bits for particular types of drilling situations. User **103** interacts through user interface **107** with database **111**. User interface includes a means for receiving a search request from user **103** and a means for providing an output to user **103** in a human-readable form. In accordance with the preferred embodiment of the present invention, the user generates a query which is defined by user-specified criteria **109** which is passed from user interface **107** to database **111**. Database **111** includes structured data which represents captured and stored organizational experience. For example, the structured data may include drilling knowledge **115** and/or drilling experience **117**. The user-specified criteria is utilized to query database ~~1~~**111** in a manner which generates as an output the relevant or analogous knowledge and experience resident or present in database **111**. This is passed through user interface **107** to user **103**. User **103** then may utilize the knowledge to make drilling decision **105**.

The user interface **107** and database **111** of **Figure 2** are preferably constructed utilizing executable program instructions. Preferably, the program instructions are executed by a general purpose data processing system, such as that depicted in **Figure 3**.

With reference now to the figures and in particular with reference to **Figure 3**, there is depicted a pictorial representative of data processing system **41** which may be programmed in accordance with the present invention. As may be seen, data processing system **41** includes processor **12** which preferably includes a graphics processor, memory device and central processor (not shown). Coupled to processor **12** is video display **14** which may be implemented utilizing either a color or monochromatic monitor, in a manner well known in the art. Also coupled to processor **12** is keyboard **16**. Keyboard **16** preferably comprises a standard computer keyboard which is coupled to the processor by means of cable **18**.

Also coupled to processor **12** is a graphical pointing device, such as mouse **20**. Mouse **20** is coupled to processor **12**, in a manner well known in the art, via cable **22**. As is shown, mouse **20** may include left button **24**, and right button **26**, each of which may be depressed, or "clicked", to provide command and control signals to data processing system **41**. While the disclosed embodiment of the present invention utilizes a mouse, ~~20~~ those skilled in the art will appreciate that any graphical pointing device such as a light pen or touch sensitive screen may be utilized to implement the method and apparatus of the present invention. Upon reference to the foregoing, those skilled in the art will appreciate that data processing system **41** may be implemented utilizing a so-called personal computer.

DESCRIPTION OF LOOM

SUBSTITUTE SPECIFICATION

LOOM is a research project in the Artificial Intelligence Research Group at the University of Southern California's Information Sciences Institute. The goal of the project is to develop and field advanced tools for knowledge representation and reasoning in Artificial Intelligence. LOOM software is the intellectual property of the University of Southern California.

The LOOM software is a language and environment for constructing intelligent applications. At the heart of LOOM is a knowledge representation system used to provide deductive support for a declarative knowledge portion of the LOOM language. Declarative knowledge in LOOM consists of definitions, rules, facts, and default rules. A deductive engine called a classifier utilizes forward-chaining, semantic unification, and object-oriented truth maintenance technologies to compile the declarative knowledge into a network designed to efficiently support on-line deductive query processing.

The LOOM system implements a logic-based pattern matcher for driving a production rule facility, and a pattern-directed method dispatching facility for supporting the definition of object-oriented methods. There is a high degree of integration between LOOM's declarative and procedural components. This permits programmers to utilize logic programming, production rule, and object-oriented programming paradigms in a single application. The LOOM software can also be used as a deductive layer that overlays an ordinary CLOS network. In this mode, users can obtain many of the benefits of using LOOM without impacting the function or performance of their CLOS-based applications.

LOOM has been distributed to more than 80 universities and corporations, and is being used in numerous DARPA-sponsored projects in planning, software engineering, and intelligent integration of information.

Licensing of LOOM:

The LOOM software is the intellectual property of the University of Southern California. It is not in the public domain; however, the University of Southern California grants permission to use this software for non-commercial purposes without a fee. If an application is covered by the terms of this fee-free license, it is not necessary to execute a written license agreement. More information about LOOM itself, its availability, and commercial licenses may be obtained from U.S.C. Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, California 90292-6695.

System Requirements:

The LOOM software requires a Common Lisp compiler to function properly. LOOM can be expected to function properly in an ANSI-compliant Common Lisp. LOOM has been tested using the following Common Lisp compilers and platforms: Macintosh Common Lisp version 2.1-4.2; Franz Allegro Common Lisp; Unix (~~Spare~~Sparc) versions 4.1-4.3, 5.0; Windows version 4xxx, 5.x (NT); Harlequin LispWorks; Unix (~~Spare~~Sparc) version 3.0.2; Windows version 4.0.1 (NT); and Lucid Common Lisp for Unix version 4.1. Although other Lisp systems may work as well, they are not known to have been tested. Newer versions of the lisp systems from these vendors should work. For example, user reports exist which indicate that LOOM works with CMU Common Lisp version 18b.

Source Code:

The LOOM software is distributed as source files which must be compiled using a Common Lisp compiler. Several versions of LOOM are available. ~~Versison~~Version 4.0 is the current release. The LOOM software is available in several formats~~.~~,

The LOOM 4.0 Source includes installation instructions, and release notes in at least the following files: Unix tar file (3.9 MB), Gzipped Unix tar file (866 kB), and Macintosh binhex file (1.4 MB).

The LOOM 3.0 Source includes installation instructions, and release notes in at least the following files: Unix tar file (3.5 MB), Gzipped Unix tar file (831 kB), and Macintosh binhex file (1.44 MB).

The LOOM 2.1 Source includes installation instructions, and release notes in at least the following files, Unix tar file (3.3 MB) and Gzipped Unix tar file (780 kB).

Documentation, Descriptive Papers, and Manuals:

1. *LOOM Reference Manual* for LOOM version 2.0, Dave Brill, December 1993. The full paper version is available on-line and in two formats as a postscript file. Note that since the *Reference Manual* is quite old, the release notes are an important additional source of documentation.

2. Preliminary *LOOM Function Card* for LOOM version 4.0. A quick summary of LOOM functions, December 1998.

3. *LOOM Function Card* for LOOM version 2.0. A quick summary of LOOM functions, December 1993. The full paper is available in postscript.

4. *Tutorial* for LOOM version 2.1. May 1995. The full paper is available in postscript.

5. LOOM *User's Guide* for LOOM version 1.4. ISX Corporation, August 1991. The full paper is available in postscript.

Additional Information:

From the Intelligent Systems Division of ISI/UCS, Los Angeles, CA, USA.

1. Useful Loom reference material:

The Intelligent Systems Division of Southern California hosts a website (isi.edu/isd) which provides LOOM reference material. Such materials include: Loom Users Guide (more structured description of language), Loom Tutorial (introductory examples and basic concepts), Loom Reference Manual (reference manual of all functions, macros, constructs, grammar, etc.).

2. Ontosaurus: From the Intelligent Systems Division of ISI/UCS, CA, USA.

THE KNOWLEDGE BASE

There are two main parts to the knowledge base: a conceptual model part, and a database part. This is analogous to a database with its schema and data parts. The conceptual part of the knowledge base is defined using concepts. There are binary concepts (otherwise known as roles or relations) and unary concepts (otherwise known as concepts or classes). A introduction to these ideas in terms of Loom can be found in the Loom reference material.

The existential database part ~~is~~ maybe easier to edit. To this end a brief summary is given of what the new instances should look like when being entered. The constructors most easily used in the database part of the knowledge base are "tell" and "~~abouf~~ about". Tell is used to assert propositions and facts about the world or domain. About references the instance those propositions refer to.

Domain Specific Conceptual Modelling and Case Capture: This is in general how a case is to be entered.

CASE: The instance below is a case instance. (It has one formation sequence name and zero or more decisions and/or observations.

______(tell (:about Case-Name
___________CASE
___________(formation-sequence Formation-Sequence-Name)
___________(decision Decision-Name)
___________(observation Observation-Name)))

The instance below is that of a formation sequence instance. It was one or more formation names.

______(tell (:about Formation-Sequence-Name
___________FORMATION_SEQUENCE
___________(formation Formation-Name)))

The instance below is that of a formation. It has one or more lithologies. It can also have start and stop depths and an interval length.

-______(tell (:about Formation-Name
___________FORMATION
___________(lithology Lithology-Name)
___________(formation-interval-length-~~f\~~ft/m NUMBER)))

The instance below is a lithology. It has one or more hardness definitions. The one below has a lithology hardness percent number meaning that there is number percent of the lithology with the hardness specified by the hardness instance in the ternary relation. Using the interval length from the formation this lithology is in, the lithology hardness amount in feet or meters can be calculated.

(tell (:about Lithology-Name

LITHOLOGY
(lithology-hardness-percent Generic-Hardness-Instance-Name NUMBER)
(lithology-type-percent NUMBER)))

The lithology definitions below has the lithology hardness amount entered directly.

(tell (:about Lithology-Name
LITHOLOGY
~~(lithology-~~ (lithology-hardness-amount-~~f/~~ft/m Generic-Hardness-Instance-Name NUMBER)
(lithology-type-amount-~~f/~~ft/m NUMBER)))

Below is a decision instance. It has goal concepts and issue concepts. It also has an action instance, an author instance, and a reasoning string. (There could be other roles too.)

(tell (:about Decision-Name
DECISION
(goal GOAL)
(issue ISSUE)
(action Action-Name)
(author Author-Name)
(authors-reasoning STRING)))

Below is a drill bit action instance. It has a drill bit role which ~~it~~is to be filled by a drill bit concept.

(tell (:about Action-Name
ACTION
(drill-bit DRILL_BIT)))

The present invention utilizes an ontological system which employs the LOOM modeling code which is available over the internet from the Information Sciences Institute. LOOM 3.0 is the

version that is currently available over the internet. The following is a description of the ontological system, and includes representative code for modeling drilling information.

THE ONTOLOGICAL SYSTEM: The system's main ontological definitions will now be outlined with both a plain English style explanation of the intended meaning and with a copy of the Loom ~~modelling~~modeling code to reference.

KNOWLEDGE MANAGEMENT CONCEPTS

This is one of the main ontological "top level concepts". Top level concepts are those that are of type thing, which is a concept defined in Loom's theory ~~BUILT~~BUILT-IN-THEORY. Any user-defined concept will be of type THING.

```
(defconcept KNOWLEDGE_MANAGEMENT_CONCEPT)
```

Knowledge management concepts can be broken down into three broad categories including I CASES, II ENVIRONMENT, III DOWNHOLE EQUIPMENT. The following is a description of each of these three categories. The types of information which is modeled in the CASES category includes date, location, decisions, issues, actions, goals, author, author's reasoning, company's reasoning, spin, observation, observation text, observation category, and observation formation. The information which is structured in the environment category includes drilling fluid concepts, rock concepts, formation sequences, single formation sequences, multiple formation sequences, formations, single ontology formations, multiple ontology formations, lithologies, and hardnesses. The types of information organized in the downhole equipment category includes bottomhole assemblies, bottomhole assembly components, and drill bits.

I. CASES

A case is (usually) a bit run case. That is, if some one experienced some bit run worthy of broadcast, then the formation sequence drilled should be represented, as well as the decisions taken on how to drill that formation sequence. That is, there are several decisions which must also be represented. A decision can be a decision about the chosen drill bit, the mud, the BHA, the flow rate, and so on. The case need not be an actual drill bit run if the person entering the case feels that he has some highly structured experience or knowledge he wishes to share. A case also has a date on which it was captured and a location which it is supposed that the case is applicable to. Representative code for case identification follows.

```
(defconcept CASE
        :is-primitive
        (:and (:exactly 1 formation-sequence)
        (:all decision DECISION)
        (:all observation OBSERVATION)))
```

(1) Date: The date is the date on which the bit run or generic knowledge was 'told' to the KB. It consists of at most 1 day, month, and/or year. Representative code follows.

```
(defconcept DATE :is-primitive
        (:and (:at-most 1 day)
                (:at-most 1 month)
                (:at-most 1 year)))
```

(2) Location: The location is the geographical position at which the bit run or knowledge is believed to be applicable to or came from. Representative code follows.

```
(defconcept LOCATION)
```

(3) Decisions: A decision here is presupposed to have several different dimensions. These include: issues, actions, goals, an author, a spin, and a reasoning. These dimensions, it is felt, provide a good balance between structure and text. The structure is to enable the formal representative and therefore search power required, and the textual for the more free flowing and readable information.

These decisions are parameter led in that you should only really have a decision object if you came to some decision about a parameter which is possible to modify in the real world and which was indeed modified with reference to some consideration (issue and goal). A parameter can be a drill bit, the flow rate, the BHA used, etc. Representative code follows.

```
(defconcept DECISION
        :is-primitive
        (:and (:exactly 1 action)
        (:at-most 10 issue)
        (:at-most 10 goal)
        (:at-most 1 authors-reasoning)
        (:at-most 1 companys-reasoning)
        (:at-most 1 author)
        (:at-most 1 spin)))
```

(4) Issues: An issue is some observation or issue that the engineer was aware of when making his decision. The issues in the current version reflect quite strongly the Best Practice Drilling database, and the link roles shown below are intended to reflect this. These could be filled with links to other media, indeed to the Notes database itself ~~though~~through URLs. Representative code follows.

```
(defconcept ISSUE
        :is-primitive
```

(:and KNOWLEDGE_MANAGEMENT_CONCEPT

(:at-most 1 symptoms-and-diagnosis-~~Iink~~link)

(:at-most 1 description-link)

(:at-most 1 parameters-link)

(:at-most 1 diagnostic-information-link)

(:at-most 1 planning-actions-~~Iink~~link)

(:at-most 1 operating-practices-~~Iink~~link)

(:at-most 1 examples-link))).

(5) Actions: An action is the real-world consequent the engineer performed as part of his decision. Representative code follows.

(defconcept ACTION

:is-primitive

(:and KNOWLEDGE_MANAGEMENT_CONCEPT

(:at-most 1 categorical-outcome)

(:at-most 1 textual-outcome)))

(6) Goals: A goal is an objective which the engineer was aiming to achieve with that decision in particular. Representative code follows.

(defconcept GOAL

""

""

:is-primitive

(:and KNOWLEDGE_MANAGEMENT_CONCEPT))

(7) Author: The author is the name of the person telling the decision to the KB. This is part of the decision and not the case as there may be a team involved, with different experts taking different

decisions, or ~~the~~there may be multiple experts involved in the one decision. Representative code follows.

```
(defconcept AUTHOR :is-primitive
        (:and (:exactly 1 first-name)
              (:exactly 1 last-name)))
```

(8) Author's Reasoning: Reasoning is a field of free-text for explanations for example of why a certain drill bit was chosen. This is to allow the possibility of incomplete, inaccurate, and even incoherent explanations for actions being stored; after all, the main reasoning or determinism for the action is the other structured information describing the circumstances in which the action was taken, such as the formation sequence. Representative code follows.

```
(defrelation authors-reasoning
        :domain DECISION
        :range STRING
        :characteristics (:closed-world :single-world»))
```

(9) Company's Reasoning: This is to allow a field which expresses the company's commonly agreed upon beliefs for the decision in question. Representative code follows.

```
(defrelation companys-reasoning
        :domain DECISION
        :range STRING
        :characteristics (:closed-world :single-world»))
```

(10) Spin: The 'spin' is the type of the knowledge captured in terms of whether it is positive or negative in its effect. That is, if a case is entered which in effect is guiding the user to not using the action used in that example, then that should be considered a type of negative knowledge. If, on the other hand, a case has as its effect an engineer retrieving a case and executing the same or similar action, then that case should be considered positive. Representative code follows.

```
(defrelation spin
        :domain DECISION
        :range SPIN
        :characteristics (:closed-world :single-world»))

(defset SPIN
        :is-primitive
        (:and KNOWLEDGE_MANAGEMENT_CONCEPT
                (:one-of 'Positive 'Negative»))))
```

(11) Observations: An observation is some state or behavior observed in the drilling process. The concept has the three roles of textual-observation, categorical-observation, and observation formation. The last of these relations allows the user to specify more accurately where the observation is valid. Observations should be left to the sort of knowledge exemplified by "ROP dips going through the sand stringers lower down in this bit run", etc. Representative code follows.

```
(defconcept OBSERVATION
""
        ""
        :is-primitive
        (:and KNOWLEDGE_MANAGEMENT_CONCEPT
                (:at-most 1 observation-formation)
                (:at-most 1 textual-observation)
                (:at-most 1 categorical-observation)))
```

(12) Observation Text: A free text field left for the user to explain himself in plain English. Representative code follows.

(13) Observation Category: This is a concept hierarchy of observations which help to

~~15~~ narrow for the purposes of search the context, subject, and object, amongst other things, of the free text.

(14) Observation Formation: This is to allow the user to more accurately contextualize his observation.

II. ENVIRONMENT

The environment here is described in terms of conceptual rock sequences. These sequences have some ontological constraints on them. For instance, if the user wishes to specify the depth and/or length of a particular section of lithology then that section has to be represented as a formation. The super-structure larger than that is the formation sequence, which can have one or more formations. Each formation can have one or more lithology. A lithology has no depth or length roles.

(1) Drilling Fluid Concepts: This is a concept representing the muds used in drilling oil wells. This is a top level concept in the ontology. Representative code follows.

```
(defconcept DRILLING_FLUID_CONCEPT)
```

(2) Rock Concepts: This is a top level concept representing all rock-related concepts, including lithologies, formations, and formation sequences. Representative code follows.

```
(defconcept ROCK_CONCEPT)
```

(3) Formation Sequences: A formation sequence is the constructor used to assemble all of the separate sub-sequences identified by the user when describing their well. The formation sequence will usually be the section the drill bit drilled in its run. A formation sequence has one or more constituent formations. Representative code follows.

```
(defconcept FORMATION_SEQUENCE
        :is-primitive
        (:and ROCK_CONCEPT
                (:at-~~least~~least 1 formation)))
```

(4) Single Formation Formation Sequences: This is a formation sequence with only the single constituent formation. Representative code follows.

```
(defconcept SINGLE_FORMATION_FORMATION_SEQUENCE
        :is
        (:and FORMATION_SEQUENCE
        (:exactly 1 formation)))
```

(5) Multiple Formation Formation Sequences: This is a formation sequence with at least two constituent formations. Representative code follows.

```
(defconcept MULTIPLE_FORMATION_FORMATION_SEQUENCE
        :is
        (:and FORMATION_SEQUENCE
        (:at-Ieast 2 tormatlonjj)
```

(6) Formations: A formation has one or more lithologies. A formation is the conceptual ~~modelling~~modeling granularity at which the user should be representing any part of his well he feels should have represented interval lengths and depths. Even stringers, or production sands, if these, it is felt, need to have depths and/or lengths attached to their storage, would need to be ~~modelled~~modeled as 'formations'. If however, it is only stringers in a majority lithology, which have unimportant positions of depth in the over-all sequence due to uncertainty or irrelevance, then these stringers are 'lithologies' and the overall stringer and majority lithology sequence is then the 'formation'. Representative code follows.

```
(defconcept FORMATION
```

```
:is-primitive
(:and ROCK_CONCEPT
(:at-Ieast 1 lithology)))
```

The relation comes-in-somewhere-after relates two formations, the first of which comes in somewhere after the other. Representative code follows.

```
(defrelation comes-in-somewhere-after
    :domain FORMATION_SEQUENCE
    :range FORMATION
    :characteristics (:multiple-valued :closed-world)
    :is (:satisfies (?formation-x ?formation-y)
        (:and
        (FORMATION ?formation-x)
        (FORMATION ?formation-y)
        (:or (comes-in-immediately-after ?formation-x ?formation-y)
            (:exists (?formation-z)
                (:and
                (FORMATION ?formation-z)
                (comes-in-somewhere-after ?formation-x ?formation-z)
                (comes-in-somewhere-after ?formation-z ?formation-
                y)))))))
```

The relation comes-in-somewhere-before relates two formations, the first of which comes in somewhere before the other. Representative code follows.

```
(defrelation comes-in-somewhere-before
    :domain FORMATION_SEQUENCE
    :range FORMATION
    :characteristics (:closed-world :multiple-valued)
    :is (:satisfies (?formation-x ?formation-y)
        (:and
```

```
(FORMATION ?formation-x)
(FORMATION ?formation-y)
(:or (comes-in-immediately-before ?formation-x ?formation-y)
    (:exists (?formation-z)
        (:and
        (FORMATION ?formation-z)
        (comes-in-somewhere-before ?formation-x ?formation-z)
        (comes-in-somewhere-before ?formation-z ?formation-
        y)))))))
```

The relation comes-in-immediately-after relates two formations, the first of which comes in immediately after the other. Representative code follows.

```
(defrelation comes-in-immediately-after
    :domain FORMATION_SEQUENCE
    :range FORMATION
    :characteristics (:closed-world :single-valued))
```

The relation comes-in-immediately-before relates two formations, the first of which comes in immediately before the other. Representative code follows.

```
(defrelation comes-in-immediately-before
    :domain FORMATION_SEQUENCE
    :range FORMATION
    :characteristics (:closed-world :single-valued))
```

(7) Single Lithology Formations: These are formations conceptualized to have only the single lithology type present (although it may have multiple harnesses). Representative code follows.

```
(defconcept SINGLE_LITHOLOGY_FORMATION
        :is
        (:and FORMATION
                (:exactly 1 lithology)))
```

(8) Multiple Lithology Formations: These are formations conceptualized to have multiple lithology types present (each of which may have multiple harnesses). Representative code follows.

```
(defconcept MULTIPLE_LITHOLOGY_FORMATION
        :is
        (:and FORMATION
        (:at-leastleast 2 lithology)))
```

(9) Lithologies: A lithology is the basic rock type, e.g., sand, shale, etc. Each lithology type also has an amount or a percentage amount for the formation in which it is present. As well as this, each lithology type is broken down into having a hardness, which then itself has an amount or a percentage amount for the lithology type. In this way, complex conceptual descriptions can be built up of the well's statistical hardness profile. It can be envisaged that a tool useful in the construction of such conceptual description could be useful; as different levels of accuracy are obtainable, from the very coarse, to the very fine, grained and, obviously, the more fine grained, the more work is required to enter the definitions. Representative code follows.

```
(defconcept LITHOLOGY
        :is-primitive
        (:and
        ROCK_CONCEPT))
```

(10) Hardnesses: There are various hardness concepts, each with their meaning specified by their UCS ranges. At the top level there is HARDNESS. Below that there are six second tier sub-

categories of hardness starting at ~~0kPSI~~0kPSI, and increasing in steps of 5kPSI. There are also finer grained hardness sub-categories below each second tier category. These start at 0 kPSI and increase in steps of 2.5kPSI. Each of the six second tier hardness categories has a LOWER_ and an UPPER_ third tier sub-category. This gives twelve third-tier hardness categories.

An interesting note to be made about harnesses is the following. If there is known to be 60% hard shale and 40% firm shale then that shale can be ~~modelled~~modeled with those stated percentages. If, however, it is only known that the shale has firm and hard harnesses in it, then a hardness FIRM_TO_HARD could be used as its hardness category (of which there is obviously 100% of). This modelling trick is useful when the exact percentage breakdown of the various constituent harnesses is unavailable. Representative code follows.

```
(defconcept HARDNESS
        :is-primitive
        (:and ROCK_CONCEPT
                (:at-most 1 ucs)))
```

Due to the fact that a lithology may have one type (e.g., it is shale), they can have more than one hardness (e.g., that shale may have 100m of very soft rock with 300m soft rock also). Representative code follows.

```
(defrelation hardness
        :domain LITHOLOGY
        :range HARDNESS
        :characteristics (:closed-world :multiple-valued))
```

The following relation calculates the amount of lithology in feet of a certain hardness. It relates the amount, the hardness and the lithology in a ternary relation.

```
(defrelation lithology-hardness-amount-ft
```

```
:arity 3
:domains (LITHOLOGY HARDNESS)
:range NUMBER
:characteristics (:closed-world :single-valued)
:is (:satisfies (?lithology ?hardness ?hardness-amount-ft)
        (:exists (?percent ?amount)
            (:and (lithology-hardness-percent ?lithology ?hardness ?percent)
                (lithology-type-amount-ft ?lithology ?amount)
                (* (/ ?percent 100.0)?amount ?hardness-amount-ft)))))
```

The following relation calculates the amount of lithology in meters of a certain hardness. It relates the amount, the hardness and the lithology in a ternary relation.

```
(defrelation Iithology-hardness-amount-m
    :arity3
~~:domains(LITHOLOGYHARDNESS)~~
    :domains(LITHOLOGY HARDNESS)
    :rangeNUMBER
    :characteristics (:closed-world :single-valued)
    :is (:satisfies (?lithology ?hardness ?hardness-amount-m)
            (:exists (?percent ?amount)
                (:and (lithology-hardness-percent ?lithology ?hardness ?percent)
~~(lithology~~              (lithology-type-amount-m ?lithology ?amount)
                    (* (/ ?percent 100.0)?amount ?hardness-amount-m)))))
```

The following relation calculates the amount of lithology in a formation in feet.

```
(defrelation ~~lithology~~lithology-type-amount-ft
~~:domainLITHOLOGY~~
    :domain LITHOLOGY
    :rangeNUMBER
    :characteristics (:closed-world :single-valued)
```

:is (:satisfies (?lithology ?amount-ft)

(:exists (?percent ?formation ?~~Iength~~length-ft)

(:and (lithology ?formation ?~~Iithology~~lithology)

(formation-interval-~~Iength~~length-ft ?formation ?~~Iength~~length-ft)

(lithology-type-percent ?~~Iithology~~lithology ?percent)

(* (/ ?percent 100.0)?~~Iength~~length-ft ?amount-ft)))))

The following relation calculates the amount of lithology in a formation in meters.

(defrelation ~~Iithology~~lithology-type-amount-m

:domain LITHOLOGY

:rangeNUMBER

:characteristics (:closed-world :single-valued)

:is (:satisfies (?lithology ?amount-m)

(:exists (?percent ?formation ?~~Iength~~length-m)

(:and (lithology ?formation ?lithology)

(formation-interval-~~Iength~~length-m ?formation ?~~Iength~~length-m)

~~(lithology~~ (lithology-type-percent ?~~Iithology~~lithology ?percent)

(* (/ ?percent 100.0) ?~~Iength~~length-m ?amount-m)))))

The following concept is an example of one of the second tier hardness concepts. It will ~~have~~have two subconcepts, upper soft and lower soft, beneath it. It is defined in terms of its ucs range. It is also ~~a~~ defined in that it is not primitive, and as such will recognize any hardness instance with its ucs range.

(defconcept SOFT

~~15~~ :is

(:andHARDNESS

(>= ucs 5000)

(≤ucs 10000)))

III. DOWN-HOLE EQUIPMENT

A down-hole tool is here defined to be anything that goes down hole and is not part of a recycling system (such as the fluid). So, a down-hole tool is a drill bit, a BHA component, and so on. Down-hole equipment concept is a top level concept.

(defconcept DOWN-HOLE_EQUIPMENT_CONCEPT)

(1) Bottom Hole Assemblies: Bottom hole assemblies have at least 1 bha component. Representative code follows.

(defconcept BOTTOM_HOLE_ASSEMBLY

:is-primitive

(:and DOWN-HOLE_EQUIPMENT_CONCEPT

(:at-~~least~~least 1 bha-component)))

(2) BHA Components: A BHA component is anything capable of being added to the BHA. So, motors, bits, MWD, VSS, under-reamers, etc, are all considered BHA_COMPONENTS. Representative code follows.

```
(defconcept BHA_COMPONENT
    :is-primitive
    (:and DOWN-HOLE_EQUIPMENT_CONCEPT))
```

(3) Drill Bits: A drill bit is a type of down-hole equipment concept with exactly 1 bit gauge. Representative code follows.

```
(defconcept DRILL_BIT :is-primitive
    (:and DOWN-HOLE_EQUIPMENT_CONCEPT
    (:exactly 1 bit-gauge)))
```

Following is a description of how one interacts with the database utilizing the LOOM artificial intelligence system. Representative code is included where necessary to explain the concepts.

HOW TO INTERACT WITH THE SYSTEM

I. Querying Instances

The functions retrieve and ask provide the interface to Loom's deductive query facility. Retrieve is used for retrieving facts (instances from a knowledge base), while ask is used to determine whether or not a proposition is true with respect to the currently stated rules and facts.

A query has one of the following forms
(retrieve ?m1 *query)*
(retrieve (?m1 ... ?mn)
(ask *query)*

The $?m_n$ are called output variables, and query is an open sentence in which the output variables appear unbound (unquantified). ~~Query~~Qquery can be any arbitrary expression in the first order predicate calculus (FOPC). The output variables must be prefixed with the character '?'.

As a note of interest, when querying for instances, the full benefits of the subsumption relation comes free. That is, if you have an instance of a drill bit called My-STR20 which is asserted (or deduced) to be of type STR20 (a type of TRICONE), when you come to look for cases with drill bits which are of type TRICONE, My-STR20 will be recognized as such by Loom's nominal query behavior. However, if a case's drill bit has been, like it is here, represented as a concept, then if you enter the query looking for cases with a drill bit of type TRICONE, Loom will, unless forced, not check the conceptual subsumption relation. Hence, to search in a more powerful manner, the function subrelations must be used, which will force Loom to check for subrelation (including subconcept) relations.

II. Examples Queries

Beginning with basic queries, a more complex query will be built.

(1) Querying Formation Sequences: This is probably the most sophisticated form of querying that will be done of the knowledge base. The two concepts likely to be of interest are formation sequences and formations (which are components of formation sequences). This is due to the more complex ternary relations used in the modelling of the lithologies (which are components of formations). It is also due to the ambiguity that will arise when users split up (or 'conceptualize') their wells into different formations (whilst, maybe, talking about essentially the same formation sequence) using different criteria. If that happens, then the user will want to look for formation sequences that, as a whole, contain, for example, 300 meters and over of very soft to soft shale type lithology. This therefore involves summing the amounts of very soft -to soft shales over all the formations constituent to each formation sequence. Hence the requirement for the sum and count functions below. Also, the query works over all sub-types of shale (e.g., balling shale). It also works over all sub-types of either soft or very soft (e.g., lower very soft). So, the lithology defined as lower very soft balling shale would also have its amount added (as per the query below) to the variable ?lithology-amount-ft.

There are several different flavors to querying formation sequences. You can query for an overall cumulative amount of a certain hardness of a certain lithology over all the formations of a formation sequence.

You can query for formations which have amounts of certain lithologies of certain hardness. The query below queries for cases which have a formation sequence which has as its constituents of its formation(s) greater than or equal to 1900 feet of very soft to

soft (including all subtypes of soft and very soft) of shale (including all sub-types of shale).

Representative code follows.

```
(retrieve ?case
        (:and
        (CASE ?case)
        (>= (sum (:collect ?lithology-amount-ft
                (:and
                (:exists (?formation-sequence ?formation ?lithology ?hardness)
                        (:and
                        (formation-sequence ?case ?formation-sequence)
                        (formation ?formation-sequence ?formation)
                        (lithology ?formation ?lithology)
                        (lithology-hardness-amount-ft ?lithology ?hardness ?~~lithology-~~lithology
amount-
ft)
                        (:or
                        (VERY_SOFT ?hardness)
                        (SOFT ?hardness))
                        (SHALE ?lithology)
~~)))))~~
                        )))))
                1900)))
```

2. Querying Authors, Dates, and Locations: The query below queries for cases that have Scott MacDonald as author, of 1998, and in the Gulf of Mexico area. Representative code follows.

```
(retrieve ?case
        (:and
        (CASE ?case)
```

```
(:exists (?decision ?date ?year ?author ?~~location~~location)
        (:and
        (date ?case ?date)
        (year ?date ?year)
        (= ?year 1998)
        (decision ?case ?decision)
        (author ?decision ?author)
        (= ?author Scott-MacDonald)
        (location ?case ?~~location~~location)
~~(= ?location GaM)))))~~
        (= ?location GOM)))))
```

3. Querying Goals: The query below queries for cases that have a drill bit decision such that one of its goals was for good ~~Rap~~ROP with good bit cleaning. Representative code follows.

```
(retrieve ?case
        (:and
        (CASE ?case)
        (:exists (?decision)
                (decision ?case ?decision)
                (DRILL_BIT_PLANNING_DECISION ?decision)
                (goal                                        ?decision
GOOD_~~Rap~~ROP_WITH_~~GOaD~~GOOD_BIT_CLEANING))))
```

4. Querying Actions: The query below queries for cases where the drill bit (planning) action was to use a drill bit of type steel-tooth. This query will work if the drill-bit role has -been filled with a concept. The second query will work if the role drill bit has been filled by an instance of a drill bit. For the querying of other actions, goals, and issues, the first type of querying would be used, as all their respective roles have their fillers in instances (of cases) represented as concepts. Representative code follows.

```
(retrieve ?case
        (:and
        (CASE ?case)
        (:exists (?decision ?action ?bit)
                (:and
                (decision ?case ?decision)
                (DRILL_BIT_PLANNING_DECISION ?decision)
                (action ?decision ?action)
                (DRILL_BIT_PLANNING_ACTION ?action)
                (drill-bit ?action ?bit)
                (subrelations STEEL-TOOTH_DRILL_BIT ?bit)))))
```

```
(retrieve ?case
        (:and
        (CASE ?case)
        (:exists (?decision ?action ?bit)
                (:and
                (decision ?case ?decision)
                (DRILL_BIT_PLANNING_DECISION ?decision)
                (action ?decision ?action)
                (DRILL_BIT_PLANNING_ACTION ?action)
                (drill-bit ?action ?bit)
                (STEEL-TOOTH_DRILL_BIT ?bit)))))
```

5. Complex ~~Quering~~Querying: The query below is an example of how many smaller queries can be stuck together to generate more complex queries. Particular note should be taken of the many individual :exists calls with their attendant parameters lists. Representative code follows.

The complex query below queries for cases that have more than or equal to 1900 feet of soft to very soft shale (as above), which had as a goal of the drill bit decision good ROP with good bit cleaning (as above), which also were recorded in 1998 in the Gulf of Mexico by Scott

MacDonald (as above), and that the drill bit chosen should be some type of ~~steel~~steel tooth drill bit.

```
(retrieve ?case
        (:and
        (CASE ?case)
        10 (>= (sum (:collect ?lithology-amount-ft
                (:and
                (:exists (?formation-sequence ?formation ?lithology ?hardness)
                        (:and
                        (formation-sequence ?case ?formation-sequence)
                        (formation ?formation-sequence ?formation)
                        (lithology ?formation ?lithology)
                        (lithology-hardness-amount-ft ?lithology ?hardness
?lithology    ?lithology-amount-ft)

                        (:or
                        (VERY_SOFT ?hardness)
                        (SOFT ?hardness))
                        (SHALE ?lithology)
)))))
                        )))))
          1900)
    (:exists (?decision ?date ?year ?author ?location)
          (:and
          (date ?case ?date)
          (year ?date ?year)
          (= ?year 1998)
```

```
        (decision ?case ?decision)
        (author ?decision ?author)
        (= ?author Scott-MacDonald)
        (location ?case ?~~location~~location)
~~(= ?location~~      (= ?location GOM)))
    (:exists (?decision ?issue)
        (:and
        (decision ?case ?decision)
        (issue ?decision ?issue)
        (subrelations ?issue GOOD_ROP_WITH_GOOD_BIT_CLEANING)
        (author ?decision Scott-MacDonald)))
    (:exists (?decision ?action ?drill-bit)
        (decision ?case ?decision)
        (DRILL_BIT_PLANNING_DECISION ?decision)
        (action ?decision ?action)
        (DRILL_BIT_PLANNING_ACTION ?action)
        (drill-bit ?action ?drill-bit)
        (subrelations STEEL-TOOTH_DRILL_BIT ?drill-bit))))
```

ABSTRACT

A method has been designed for storing drilling knowledge and experience in a highly structured fashion that permits the user to identify drilling cases that meet user-specified criteria and to retrieve the knowledge and experience relating to those cases. In this way the user is able to retrieve the knowledge and experience learned in cases that are analogous to one or more current cases they are studying.

SUBSTITUTE SPECIFICATION

Docket No. **9001RF-34875**

DECLARATION FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe that I am an original, first, and joint inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled

CASE-BASED DRILLING KNOWLEDGE MANAGEMENT SYSTEM

further identified by attorney docket no. **9001RF-34875.**

This application claims the benefit of U.S. Provisional Application Serial No. **60/140,119,** filed **18 JUNE 1999,** entitled **Case-Based Drilling Knowledge Management System.**

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the Office all information known to my person to be material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, Sec. 1.56(a).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

I hereby appoint **Melvin A. Hunn, Reg. No. 32,574** and **Kenneth C. Hill, Reg. No. 29,650** to prosecute this application and to transact all business in the U.S. Patent and Trademark Office in connection therewith.

Exhibit A

Please send all correspondence to:

Melvin A. Hunn
Registration No. 32,574
HILL & HUNN LLP
201 Main Street, Suite 1440
Fort Worth, Texas 76102-3105
817/332.2113

Inventor's Signature: ______________________

Full Name of Inventor: **Alan Nicholas Flett**

Date of Signature: 21/8/2000

Residence and Post Office Address:
8 Mill Lade Wynd, Danestone, Aberdeen AB22 8QN, Scotland

Citizenship: United Kingdom

Inventor's Signature: ______________________

Full Name of Inventor: **Derek H. Sleeman**

Date of Signature: 3 Sept 2000

Residence and Post Office Address:
10 The Chanonry, Aberdeen AB24 1RN, Scotland

Citizenship: United Kingdom

Inventor's Signature: ______________________

Full Name of Inventor: **Alun D. Preece**

Date of Signature: 4/9/2000

Residence and Post Office Address:
10 Fare Park Drive, Westhill, Aberdeenshire AB32 6WE, Scotland

Citizenship: United Kingdom

Electronic Patent Application Fee Transmittal

Application Number:	09595550
Filing Date:	16-Jun-2000
Title of Invention:	CASE-BASED DRILLING KNOWLEDGE MANAGEMENT SYSTEM
First Named Inventor/Applicant Name:	Alan Nicholas Fleet
Filer:	James E. Bradley/Kimberly Tobola
Attorney Docket Number:	9001RF-34875

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Petition-revive unintent. abandoned appl	1453	1	1620	1620
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
		Total in USD ($)		**1620**

Electronic Acknowledgement Receipt

EFS ID:	10642103
Application Number:	09595550
International Application Number:	
Confirmation Number:	1122
Title of Invention:	CASE-BASED DRILLING KNOWLEDGE MANAGEMENT SYSTEM
First Named Inventor/Applicant Name:	Alan Nicholas Fleet
Correspondence Address:	Melvin A Hunn Hill & Hunn LLP 201 Main Street Suite 1440 Fort Worth TX 76102-3105 US 817/332.2113 -
Filer:	James E. Bradley/Kimberly Tobola
Filer Authorized By:	James E. Bradley
Attorney Docket Number:	9001RF-34875
Receipt Date:	01-AUG-2011
Filing Date:	16-JUN-2000
Time Stamp:	17:43:15
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Credit Card
Payment was successfully received in RAM	$1620

RAM confirmation Number	4645
Deposit Account	500259
Authorized User	BRADLEY,JAMES

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Petition for review by the Office of Petitions.	Pet.pdf	321929 ff6b80a1043a1e886fa543a563ddba7fe59aa62b	no	4
Warnings:					
Information:					
2	Miscellaneous Incoming Letter	PetExhA.pdf	320932 d9dd217d37f49a075c945ff7e99207bde28ac7a5	no	3
Warnings:					
Information:					
3	Miscellaneous Incoming Letter	PetExhB.pdf	624117 b485cfd9534e123402c88e5e1bcf2ea0c33cc1eb	no	3
Warnings:					
Information:					
4	Miscellaneous Incoming Letter	PetExhC.pdf	1903178 4f184677efe0990a9525661b03bf4717ea9d8e6f	no	7
Warnings:					
Information:					
5	Oath or Declaration filed	PetExhD.pdf	241740 d23925e50a39e4b0cfcbb7f8d230a787b56ef90e	no	2
Warnings:					
Information:					
6	Assignee showing of ownership per 37 CFR 3.73(b).	373b.pdf	83762 50208ef106720bafe0d57bd7e00cef980163cb2e	no	1
Warnings:					
Information:					

7	Power of Attorney	POA.pdf	90797 e753587ce84cfa8d163a4e15b242fd29040721b6	no	1
Warnings:					
Information:					
8	Amendment after Notice of Allowance (Rule 312)	Resp.pdf	161220 519001e118b53b73fc9eb828fe743898b6d54e8d	no	2
Warnings:					
Information:					
9	Abstract	Abstract.pdf	40210 6a789e887038d814a70c12298c285ead9439177e	no	1
Warnings:					
Information:					
10	Specification	CleanSpec.pdf	2654381 82d4d119e618809774f472df1c67626d03416c22	no	32
Warnings:					
Information:					
11	Specification	MarkedupSpec.pdf	2937898 0cabbdcd87800469fdce19662d18814499bfc301	no	35
Warnings:					
Information:					
12	Oath or Declaration filed	RespExhA.pdf	242231 672dcc353d737852d8dd0b41a1f99c79c358c36c	no	2
Warnings:					
Information:					
13	Fee Worksheet (SB06)	fee-info.pdf	30706 7c1c9dbab93a84c79a2298278d803e18f1ce1201	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	9653101	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.